SIEBEL eBusiness 7

UPGRADE GUIDE FOR UNIX

VERSION 7.5, REV. A

JANUARY 2003

12-DX9B8J

Siebel Systems, Inc., 2207 Bridgepointe Parkway, San Mateo, CA 94404

Printed in the United States of America

Contents

Introduction

Chapter 1. Planning an Upgrade

Chapter 2. Data Migration Considerations

Chapter 3. Preupgrade Tasks

Chapter 4. Upgrading the Siebel eBusiness Application

Chapter 5. Postupgrade Tasks

Chapter 6. Migrating the Client Configuration to the Web Client

Appendix A.Upgrade Planning Worksheet

Appendix B.Upgrading Without a Development Environment

Appendix C.Error Messages

Index

Introduction

This guide contains guidelines and specific instructions for upgrading your Siebel eBusiness Applications to the latest release, Siebel 7.

This book will be useful primarily to people whose titles or job descriptions match one of the following:

Siebel System Administrators	Persons responsible for administering the whole system, including installing, maintaining, and upgrading Siebel products.
UNIX System Administrators	Persons with root-level access who are responsible for UNIX system administration.
Database Administrators	Persons who administer the database system, including data loading; system monitoring, backup, and recovery; space allocation and sizing; and user account management.

How This Guide Is Organized

This guide provides required information for upgrading Siebel eBusiness Applications. To make the best use of this guide, carefully read Chapter 1, “Planning an Upgrade” and Chapter 2, “Data Migration Considerations,” first to understand the changes that occur during the Release 7.5 upgrade.

To successfully upgrade your Siebel eBusiness Applications, you will conduct the upgrade of your development environment and then repeat the process for your production environment. Table 1 on page 11 displays the sequence of chapters you will use to upgrade your development and production environments.

Because preupgrade steps are similar for both development and production environments, they are included in one chapter: “Preupgrade Tasks.” Likewise, upgrade steps are included in one chapter, “Upgrading the Siebel eBusiness Application,” and postupgrade steps are included in one chapter: “Postupgrade Tasks.” Because you will use these chapters to upgrade both your development and production environments, you will not proceed sequentially through the chapters in this book. Instead, you will use the chapters in the sequence indicated in Table 1 on page 11.

Table 1 displays the sequence of chapters you will use to upgrade your development and production environments.

Table 1. How to Use this Guide to Conduct Your Upgrade

Sequence	Chapter Title
Upgrading the Development Environment	
1	"Introduction"
2	Chapter 1, "Planning an Upgrade"
3	Chapter 2, "Data Migration Considerations"
4	Chapter 3, "Preupgrade Tasks"
5	Chapter 4, "Upgrading the Siebel eBusiness Application"
6	Chapter 5, "Postupgrade Tasks"
7	Chapter 6, "Migrating the Client Configuration to the Web Client"
Upgrading the Production Environment	
8	Chapter 3, "Preupgrade Tasks"
9	Chapter 4, "Upgrading the Siebel eBusiness Application"
10	Chapter 5, "Postupgrade Tasks"
11	Chapter 6, "Migrating the Client Configuration to the Web Client" **NOTE.** You will only use this chapter as part of your production environment upgrade if you did not perform client configuration migration as part of your development environment upgrade.

NOTE: If you encounter errors during the upgrade process, refer to Appendix C, "Error Messages," which contains information about some of the most common error messages.

Naming and Typographical Conventions in This Guide

This guide follows several naming conventions:

- Release 7.5 refers to the current release of the Siebel eBusiness Applications.
- Release 7.0.x refers collectively to all versions of Siebel 7.0 that are supported in upgrade to Release 7.5; for example, Release 7.0.3 or Release 7.0.4. Refer to the Upgrade section in *Siebel System Requirements and Supported Platforms* on Siebel SupportWeb at http://ebusiness.siebel.com/supportweb/ for specific information on which release numbers are meant by Release 7.0.x.
- Release 7.x refers collectively to all versions of Siebel 7; for example, Release 7.0.3, Release 7.0.4, or Release 7.5. Refer to the Upgrade section in *Siebel System Requirements and Supported Platforms* on Siebel SupportWeb at http://ebusiness.siebel.com/supportweb/ for specific information on which release numbers are meant by Release 7.x.
- Release 6.x refers collectively to all versions of Siebel 6 that are supported in upgrade to Release 7.5.
- Release 5.x refers collectively to all versions of Siebel 5 that are supported in upgrade to Release 7.5.
- The term “Windows” refers to all Microsoft Windows operating systems listed as supported for this release in *Siebel System Requirements and Supported Platforms*.
- *`SIEBEL_ROOT`* refers to the directory where you have installed the Siebel eBusiness Application; for example, `c:\sea7xx`.
- *`DBSRVR_ROOT`* refers to the directory where you have installed the Siebel Database Server.
- *`CLIENT_ROOT`* refers to the directory where the Siebel Dedicated Web Client is installed.

Unless otherwise noted, all references to Siebel eBusiness Applications documentation refer to the manuals provided with your Release 7.5 software.

Siebel Servers run in both Windows and UNIX environments. Because all versions of the UNIX operating system are case-sensitive, if you are running your Siebel eBusiness Applications under UNIX, treat all filenames, directory names, pathnames, parameters, flags, and command-line commands as lowercase, unless you are instructed otherwise in the product.

If your deployment currently runs under Windows, but you may switch to a UNIX environment, or deploy UNIX servers in the future, it is a good idea to follow the same practice, which avoids the need to rename everything later.

NOTE: Your Siebel implementation may not have all the features described in this guide, depending on which software modules you have purchased.

Revision History

Upgrade Guide for UNIX, Version 7.5, Rev. A

Topic	Revision
“Overview of the Upgrade Process”	Revised this topic. Added new graphics of the upgrade process.
Data Migration Considerations for Release 7.x Applications	Removed this topic. This information is now available as a technical note on Siebel SupportWeb.
“Tables Modified During Upgrade to Release 7.5.2”	New topic.
“Tables Seeded During Upgrade to Release 7.5.2”	New topic.
“Preupgrade Tasks”	Revised checklist of preupgrade tasks. Checklist and chapter now contain preupgrade tasks for both development and production environments.
“Verifying Sort Order on Oracle Client”	Revised content to clarify code page requirements.
“Updating Statistics on DB2 UDB”	New topic.
“Checking In Archived Release 5.x and Release 6.x Repository Objects”	New topic.
“Continuing with the Upgrade”	New topic.
“Upgrading the Siebel eBusiness Application”	New graphic. Revised checklist of upgrade tasks. Checklist and chapter now contain upgrade tasks for both the development and the production environment.
“Migrate Strings, Merge Labels and Fields, and Merge Templates”	New table that includes information about two files: labelmerge.txt and templatemerge.txt.
“Continuing with the Upgrade”	New topic.
“Postupgrade Tasks”	New graphic. Revised checklist of postupgrade tasks. Checklist and chapter now contain postupgrade tasks for both the development and the production environment.

Topic	Revision
“Migrating Custom Columns on Obsolete Tables”	Revised this topic. Added information about using the script `rpt_obs_tbl_custom_cols.sql`.
“Running Update Statements”	New topic.
“Reapplying Custom Extensions and FK Columns on Obsolete Tables”	New information about FK columns.
“Migrating Custom Business Components and Resolving Conflicts”	Revised this topic.
Validating the Upgrade	Removed this topic. Renamed information that used to be in this topic about using the DICTUTL utility to “Validating Dock Objects and Rule Definitions.”
“Validating Dock Objects and Rule Definitions”	Revised this topic.
“Continuing with the Upgrade”	New topic.
Appendix B, “Upgrading Without a Development Environment”	New appendix.
Appendix C, “Error Messages”	New appendix. It contains some common errors that may occur during the upgrade. It does not include all possible error messages.

Planning an Upgrade 1

This chapter provides instructions for upgrading to Release 7.5 from several prior releases of Siebel eBusiness Applications on Siebel Systems supported database platforms. While the general upgrade procedure is the same for all releases and platforms, the exact steps may vary slightly depending on the database platform and release from which you are upgrading.

Where the upgrade process differs from a platform standpoint, the proper steps for each release and database platform are clearly identified within this guide. For information about supported upgrade paths, see *Siebel System Requirements and Supported Platforms* on Siebel SupportWeb at http://ebusiness.siebel.com/supportweb/.

A successful upgrade requires:

- Familiarity with the basic conventions of the UNIX operating system.
- Familiarity with your specific Siebel eBusiness Applications deployment.
- Expertise in network connectivity, disk and file sharing, and software installation on your chosen application server and client operating systems.
- User accounts with root access on UNIX to install new software.
- Expertise in database installation, tuning, and administration in your chosen relational database management system (RDBMS).
- Familiarity with the Siebel Packager Utility. For more information about the Siebel Packager Utility, see *Siebel Web Client Administration Guide*.
- Detailed understanding of customizations to support previous implementation of Siebel eBusiness Applications.

This document explains how to upgrade your Siebel eBusiness Applications on several different database platforms, operating system platforms, and application server platforms. However, certain database and operating system platforms may not have been supported in previous releases, or may have been discontinued in the current release. Certain combinations of database and operating system platforms may also not be supported.

For this reason, information may appear in this manual about one or more platforms or combinations of database platforms and operating systems that are not currently available. For a list of currently supported systems, see *Siebel System Requirements and Supported Platforms* on Siebel SupportWeb at http://ebusiness.siebel.com/supportweb/.

Overview of the Upgrade Process

The upgrade process is composed of three distinct phases that you should perform in sequence, as illustrated in Figure 1.

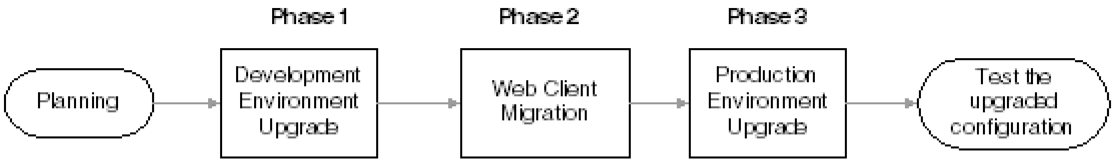

Phase 1
Phase 2
Phase 3
Planning
Development Environment Upgrade
Web Client Migration
Production Environment Upgrade
Test the upgraded configuration

Figure 1. The Upgrade Process

- **Phase 1: Development Environment Upgrade.** Phase 1 includes upgrading your development environment and performing a repository merge to produce a new Release 7.5 custom configuration. During this phase, your existing configuration and development database are upgraded to Release 7.5.

  You should upgrade a development environment before you upgrade a production environment in order to produce and test a Release 7.5 configuration containing your customizations. Then, you should thoroughly test and validate that configuration before you proceed with the production environment upgrade.

  To complete Phase 1, follow the instructions to upgrade a development environment in Chapter 3, “Preupgrade Tasks,” Chapter 4, “Upgrading the Siebel eBusiness Application,” and Chapter 5, “Postupgrade Tasks.”

- **Phase 2: Migrating the Client Configuration to the Web Client.** Phase 2 includes migration of your client configuration from Release 6.x or Release 7.0.x Windows client to the Release 7.5 Web client.

  To complete Phase 2, follow the instructions in Chapter 6, “Migrating the Client Configuration to the Web Client.”

  NOTE: If you are upgrading from Release 7.0.x to Release 7.5, (for example, from Release 7.0.4 to Release 7.5), you do not need to perform Phase 2.

- **Phase 3: Upgrading the Production Environment.** Phase 3 includes upgrading your production Siebel Database, Siebel Servers, and File Server or File System. During this phase, you will distribute the new configuration to your production Siebel clients.

  To complete Phase 3, follow instructions to upgrade a production environment in Chapter 3, “Preupgrade Tasks,” Chapter 4, “Upgrading the Siebel eBusiness Application,” and Chapter 5, “Postupgrade Tasks.”

Figure 2 illustrates the high-level steps for a development environment upgrade.

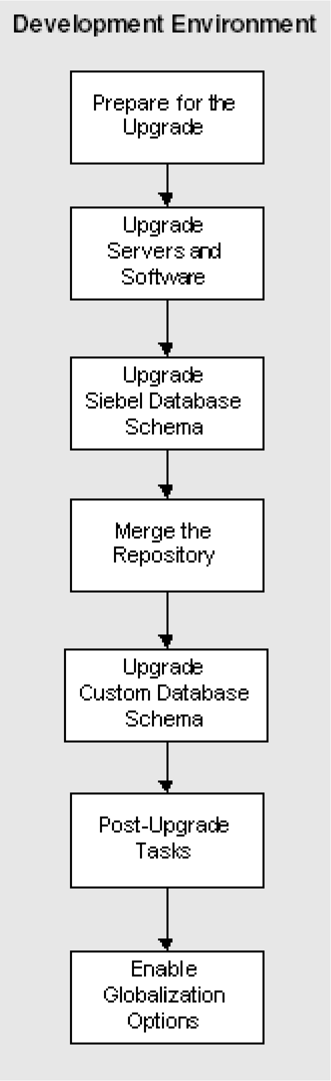

Development Environment
Prepare for the Upgrade
Upgrade Servers and Software
Upgrade Siebel Database Schema
Merge the Repository
Upgrade Custom Database Schema
Post-Upgrade Tasks
Enable Globalization Options

Figure 2. Upgrade of the Development Environment

Figure 3 illustrates the high-level steps for a production environment upgrade.

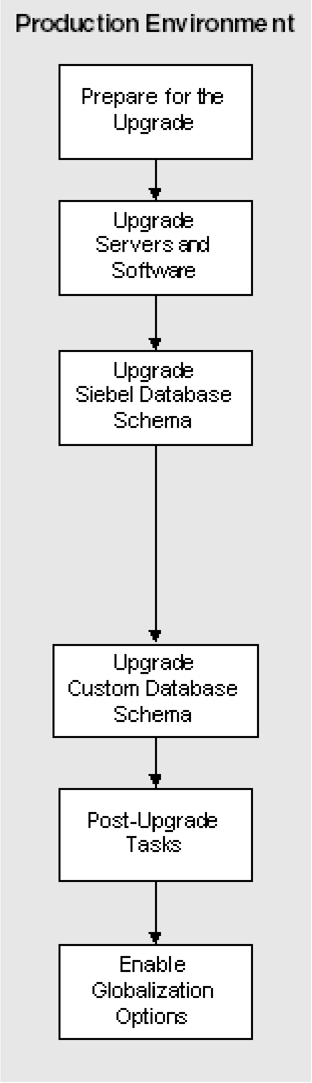

Production Environment
Prepare for the Upgrade
Upgrade Servers and Software
Upgrade Siebel Database Schema
Upgrade Custom Database Schema
Post-Upgrade Tasks
Enable Globalization Options

Figure 3. Upgrade of the Production Environment

Supported Upgrade Paths

This guide does not contain information about supported upgrade paths. For information about supported upgrade paths, including third-party software and hardware, see *Siebel System Requirements and Supported Platforms* on Siebel SupportWeb at http://ebusiness.siebel.com/supportweb/.

Nonstandard Upgrades and Migrations

Nonstandard upgrades and migrations are outside the scope of this guide. Given the complexity of a nonstandard upgrade or migration, contact Siebel Expert Services for assistance. Nonstandard upgrades and migrations include:

- Upgrades from one database platform to another.
- Upgrades from one primary language to another.
- Migrations to Unicode.
- Upgrades from one operating system to another.
- Migrations between Siebel eBusiness Industry Solutions or Siebel Financial Services and Siebel eBusiness Applications.
- Upgrades from Release 6.x involving Siebel Connector for SAP, R/3.

NOTE: If your enterprise uses SAP and you have implemented Siebel System's Application Integration (EAI) product, refer to *Siebel Connector for SAP R/3* for upgrade instructions.

Upgrades from Release 7.0.x to Release 7.5

Your upgrade procedure is almost identical to the other supported upgrade paths to Release 7.5, except for several minor, but important variations. Variations are identified at the beginning of each relevant procedure. Carefully read the documentation and perform every procedure in this guide as instructed. To avoid errors in your upgrade, pay close attention to cautions and notes (shown in the examples below) whenever they appear in this guide.

- **Steps to skip.** Notes, such as the one shown in the following example, alert you that you can skip a step under certain circumstances.

  NOTE: Skip this procedure if you are performing an upgrade from Release 7.0.x to Release 7.5.

- **Additional steps to perform.** Cautions, such as the one shown in the following example, alert you that you must perform additional steps in certain circumstances.

  CAUTION: If you previously upgraded to Release 7.0.x and you already ran the UTC conversion utility, perform the following steps to upgrade the delta columns (those that were not UTC-enabled in 7.0.x but are UTC-enabled in 7.5).

- **Steps with replacement steps.** Notes, such as the one shown in the following example, explain the substitution of replacement steps, where applicable.

  NOTE: Skip this step if you are upgrading from Release 7.0.x to Release 7.5, because you already created your 32-KB tablespace and bufferpool. Instead, perform the task described in “Increasing DB2 32-KB Tablespace” on page 154.

The instructions that are different for your upgrade from Release 7.0.x to Release 7.5 are summarized below.

- You may skip the following procedures, since you already completed them during your upgrade to Release 7.0.x:
    - Data migration procedures in the preupgrade and postupgrade chapters. (See "Preparing Application Data for Upgrade" and "Additional Postupgrade Tasks for Specific Applications.")
    - If your RDBMS is IBM DB2 UDB: Installing stored procedures and user-defined functions. (See "Installing the Stored Procedures and User-Defined Functions.")
    - If your RDBMS is IBM DB2 UDB: Creating 32-K tablespace and bufferpool. (See "Creating DB2 32-KB Tablespace and Bufferpool" for both development and production environments.)
    - Identify and resolve duplicates in your Siebel database. (See "Identifying and Resolving Duplicates.")
    - Repository Preparation Wizard to prepare the prior customer repository for the merge. (See "Migrate Strings, Merge Labels and Fields, and Merge Templates.")
    - Migrating client configurations to the Siebel Web Client. (See Chapter 6, "Migrating the Client Configuration to the Web Client.")
    - Reapplying custom extensions and FK columns on obsolete tables. (See "Reapplying Custom Extensions and FK Columns on Obsolete Tables.")
- You may need to perform the following additional procedures:
    - **If your RDBMS is IBM DB2 UDB.** Increasing 32-K tablespace. See "Increasing DB2 32-KB Tablespace."
    - **If your deployment is enabled for global time zone support.** If you previously upgraded to Release 7.0.x and you already ran the UTC conversion utility, you need to upgrade the delta columns (those that were not UTC-enabled in 7.0.x, but are UTC-enabled in 7.5). See "Setting Up Your Environment to Support Global Time Zone."

- **If you made customizations to Release 7.0.x seeded workflows.** During an upgrade, seeded workflows that were shipped with Release 7.0.x will be replaced by seeded workflows that are shipped with Release 7.5. Customizations will be preserved and migrated, but you will need to manually reimplement them in order for them to work properly after the upgrade. See “Siebel Workflow” on page 242.

Before You Start

Before you begin your Siebel upgrade, complete the following steps:

1 Gather the username, password, and system information that you will need during the upgrade and complete the Upgrade Planning Worksheet, located in Appendix A, "Upgrade Planning Worksheet."

2 Carefully read the chapters in this guide that correspond to your upgrade path to understand the complete upgrade process for your operating system and RDBMS platform combination. Upgrade paths are listed in *Siebel System Requirements and Supported Platforms* on Siebel SupportWeb at http://ebusiness.siebel.com/supportweb/ .

3 Read *Siebel System Requirements and Supported Platforms* to be sure you know the supported computer and operating system platforms and third-party programs for this release of your Siebel eBusiness Applications.

4 Schedule database and system administrator assistance.

Upgrading is a database-intensive process. For this reason you should have an experienced database administrator available to assist during your upgrade. Upgrading also requires the installation of new Siebel and third-party software on your Siebel Servers.

5 Carefully read the relevant chapters of *Siebel Server Installation Guide* for the operating system you are using.

6 Read *Release Notes* and *Siebel Alerts* documentation on Siebel SupportWeb for new information that this guide does not contain.

7 Prepare a comprehensive upgrade schedule that includes a time line for completing specific preupgrade and postupgrade tasks, including thorough testing of your development environment after you upgrade your development environment and before you upgrade your production environment.

Useful Resources

Your Siebel implementation team performs a number of actions to install and implement Siebel eBusiness Applications that are described in several Siebel publications:

- The current guide for upgrade and configuration instructions for Siebel eBusiness Applications.
- The *Siebel Server Installation Guide* for your operating system for server and RDBMS installation and configuration instructions.
- *Siebel Web Client Administration Guide* for client installation and configuration instructions.
- *Siebel System Requirements and Supported Platforms* available on Siebel SupportWeb at http://ebusiness.siebel.com/supportweb/ for supported computer and operating system platforms and supported third-party programs for this release of your Siebel eBusiness Applications.
- *Release Notes* and *Siebel Alerts* documentation on SupportWeb for late-breaking information that the current guide does not yet contain.
- *Siebel Server Administration Guide* for details on how to administer, maintain, and expand your Siebel servers.
- If you have a license for Siebel Tools, you will find information about configuring Siebel eBusiness Applications in *Siebel Tools Reference*.
- *Siebel Remote and Replication Manager Administration Guide* as a resource for Siebel Remote functionality.

Data Migration Considerations 2

In Release 7.x, changes to existing functionality and the structure of Siebel eBusiness Applications have required significant changes to the Siebel Data Model. During the Siebel Database upgrade, data held in Siebel Database tables is moved to accommodate this new table structure. This chapter provides information about how and where this data movement occurs so that you can better plan the upgrade of specific modules.

Changes to the Siebel Data Model

This section describes data model changes for Release 7.x including access control, categorization, and the party model. Use this information to plan your upgrade.

Access Control

Access control refers to all mechanisms to control visibility within Siebel eBusiness Applications. This includes but is not limited to positions, responsibilities, organizations, and access groups. To implement access control within your Siebel eBusiness Applications deployment, your Siebel Administrator creates relationships between people and resources (a more general term for data that includes views and functionality). These relationships or policy are authorizations. Both people and resources can be grouped and placed in hierarchies to ease the administration.

Deployment of Siebel eBusiness Applications over the Web provides external users, such as customers and channel partners with varying access levels, the ability to directly access data and application functionality. This change introduces a new set of data and content access dynamics:

- An exponential increase in the amount of content that will be distributed by the Siebel eBusiness Applications including Master data (data that is static and referential, such as Products) and Customer data (data that is created and managed by users of applications, such as Opportunities).
- An exponential increase in the number of users and entities that will access the data and added complexity of relationships between users (partners, competitors, browsers, customers).
- A significant increase in the complexity of access control policies (one data item or group of data items can be accessed by one or many users or groups, but not by all).

To support access control, data migration occurs during the upgrade to Release 7.5. The following two sections describe this migration.

Multi-Org Visibility for S_CONTACT and S_ORG_GROUP

Before Release 7.5, a Person or Household could not be made visible to multiple Business Units (Organizations). With the introduction of S_CONTACT_BU and S_ORG_GROUP_BU, the application can now share People and Households across multiple Business Units.

The upgrade to 7.5 populates both the S_CONTACT_BU and S_ORG_GROUP_BU tables with one record for each record in the S_CONTACT and S_ORG_GROUP tables. After the upgrade, Contacts and Households continue to be visible from the Business Unit they belonged to prior to the upgrade.

Access Group and Userlist

For 7.5, two new Siebel Extension tables were added to the S_PARTY, S_PARTY_GROUP and S_USERLIST tables to hold Access Group and User List attributes, respectively. The upgrade adds records to the S_PARTY_GROUP and S_USERLIST tables for existing S_PARTY Access Group and User List records. Multi-Org visibility is supported for the new tables, so the upgrade also adds corresponding intersection table records to the S_PARTY_GRP_BU and S_USERLIST_BU tables.

NOTE: For more information about access control, see *Security Guide for Siebel eBusiness Applications*.

Categorization

Categorizing content has major benefits: it simplifies access control policy design and management as System Administrators can specify access on an aggregate set of master data items. As a result, content becomes more easily searchable and accessible to users through navigation. Having intuitively navigated to one item, users will likely find many related items of potential interest.

Product categorization was available in the Siebel 6.x data model. However, Release 7.x supports categorization of additional data structures such as auction items and literature items.

Data Migration to Support Categorization

In Siebel 6.x, categories could be shared across multiple catalogs and could have multiple parents. In Release 7.x, a category can only belong to one catalog and have at most one parent catalog. To accommodate the new structure, the database upgrade generates copies of categories and category hierarchies that were previously shared across multiple catalogs.

For example, the following catalog and category hierarchy:

```
                          CATd
CTLG1            CATb
        CATa              CATe
CTLG2            CATc
                          CATf
```

results in separate hierarchies of copied categories after the upgrade:

```
                         CATd
                 CATb
                         CATe1
CTLG1    CATa
                         CATe
                 CATc
                         CATf


                         CATd
                 CATb
                         CATe1
CTLG2    CATa
                         CATe
                 CATc
                         CATf
```

NOTE: For more information about categorization, see *Security Guide for Siebel eBusiness Applications*.

Party Model

Release 7.x introduces the concept of a party table (S_PARTY), in which all persons and organizational units are held. With the new model, Accounts, Organizations, Internal Divisions, Contacts, Employees, Positions, and Households are all considered parties and can be referenced from the same table, S_PARTY.

Most of the tables that housed these data entities still exist and are still used, but they are now extension tables to the new base table. Data is brought into the business components through an implicit join.

Additionally, Release 7.x now uses a single-person table and a single-organization unit table. For example, Employees and Contacts are now combined in the same table (S_CONTACT). Similarly, internal and external Organization Units are now combined in the table (S_ORG_EXT).

The new S_PARTY table is the primary table in the Party or Single-Person model and the base table for all Party business components. Siebel extension tables: S_USER, S_EMPLOYEE, S_CONTACT, S_ORG_EXT, S_POSTN, and S_BU support the Party model. Each non-person party directly or indirectly has person members, such as employees or contacts.

The new Party model has made several tables obsolete. The S_EMPLOYEE table is obsolete as its functionality was merged into the S_CONTACT table. The S_ORG_INT table is obsolete, as its functionality has been merged into S_ORG_EXT. S_EMP_POSTN has been replaced by S_PARTY_PER.

There are several other new tables in the data model supporting the consolidation of S_EMPLOYEE with S_CONTACT and of S_ORG_INT with S_ORG_EXT. The S_USER table stores Siebel User information. The S_EMP_PER stores attributes for Brand-Owner Employees and Partner Users who are considered agents of the Brand-Owner, supporting sales and service activities on behalf of the Brand-Owner. The S_BU table stores Organization information.

As a result of the party model, the tables listed in Table 2 are obsolete in Release 7.x:

Table 2. Obsolete Tables

Obsolete Tables in Siebel 7
S_EMPLOYEE
S_EMPLOYEE_ATT
S_ORG_INT
S_ORG_INT_ATT
S_POSTN_RPT_REL

Figure 4 depicts the new Party changes to the data model:

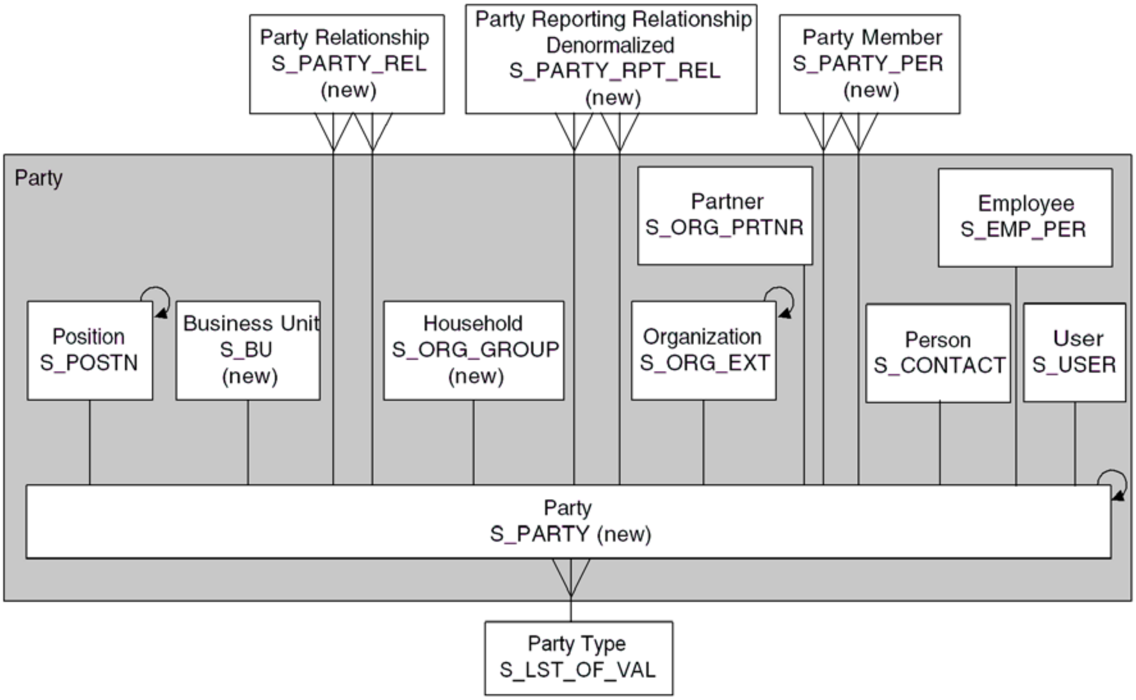


Figure 4. Party Model

Given this new table structure and new business object definitions, it is important to understand how the Release 7.x upgrade works. The following steps occur during the Release 7.x upgrade:

Data Migration to Support Party Model

- Migration of data from S_EMPLOYEE to S_CONTACT, S_USER, S_EMP_PER for standard Siebel columns
- Migration of data from S_ORG_INT to S_ORG_EXT, S_BU for standard Siebel columns
- Creation of the Party table records for each previous contact, position, employee, account, division

Business Component Definitions

- Business component definitions updated to reference S_PARTY as Primary Table (for example, Employee, Contact, Position, Account)
- Standard and Custom joins to S_EMPLOYEE changed to S_CONTACT, S_USER, S_EMP_PER
- Standard and Custom joins to S_ORG_INT changed to S_ORG_EXT
- Implicit joins set for custom fields created on business components that have been retargeted to S_PARTY. For example, if a custom field, Alternate Phone, existed on the Contact business component, the upgrade would initiate the following actions:
    - Contact business component would be retargeted to S_PARTY
    - Join to S_CONTACT from S_PARTY defined on Contact business component
    - Implicit join set for the Alternate Phone field

Additional Considerations

With the Access Control mechanism comes two new visibility mode types, catalog visibility and group visibility. For certain standard Siebel eBusiness Application configurations, the view mode and picklist mode have catalog visibility and will only render data visible if the catalogs, categories, and access groups are defined.

As will be discussed later, one of the considerations during the upgrade will be to move forward with the access control model and carefully plan and prepare the access groups, catalogs, categories as part of the upgrade phase.

Alternatively, if continued use of Multi-Org is desired, changes to default view and picklist configuration will need to be modified as part of the upgrade.

NOTE: Make sure data is visible after the production upgrade.

For more details on the Release 7.x data model as it relates to the Party model, please see *Siebel eBusiness Data Model Reference*. For more information on the new Access Control and Security mechanism in Release 7.x, see *Security Guide for Siebel eBusiness Applications*.

Columns Added to the S_SRC_PAYMENT Table

If your organization exposes the S_SRC_PAYMENT table to any of its Siebel eBusiness Applications, be aware that upgrading to Release 7.x introduces a new, required column, called TYPE_CD, to this table. This column has a default value of Payment, indicating that this is a payment from your organization to an external organization. For example, this might denote a payment of marketing funds to your customer.

If your organization has used the S_SRC_PAYMENT table to store other types of payments (particularly, payments from your customers to your organization), you will want to reset the parameter in this column to fit your data, and then run Siebel System's Enterprise Integration Management (EIM) product. This will update the table and allow you to review your results before migrating to your production environment.

The valid values for this column as shipped with Siebel eBusiness Applications are:

- Payment
- Receipt
- Write–Off
- Adjustment

For information on how to update the Lists of Values, refer to *Siebel Tools Reference*.

HTML Attachments Migrated to Content Base Tables

In Release 7.5, HTML Attachments (which were introduced in Siebel 7) are upgraded into Content Base tables. The Content Base system is designed to support applications requiring content management by allowing content to be stored in the database or in the Siebel file system. The system has the ability to store content of various types including plain text, markup languages (for example, SGML and XML), and image files (for example, GIF and JPEG) by breaking it up and then rendering the content when required.

The upgrade from Release 7.0.x to Release 7.5 migrates data from the S_WEB_CNTNT table to the S_CB_CNTNT_SET, S_CB_ASSET, and other Content Base tables.

Data Migration Considerations for Release 7.x Applications

For information about application-specific data migration, see Siebel SupportWeb for a technical note.

Tables Modified During Upgrade to Release 7.5.2

This section identifies Siebel tables that are modified during an upgrade.

- For upgrades from Release 6.x, see Table 3 on page 40.
- For upgrades from Release 7.0.x, see Table 4 on page 56.
- For upgrades from Release 7.5.1, see Table 5 on page 72.

Modifications may consist of inserted records (I), updated records (U), or referenced records (R); for example,

- S_EVT_ACT has one or more inserted records (I).

  ```
  insert into S_EVT_ACT...
  ```

- S_CONTACT has one or more updated records (U).

  ```
  UPDATE S_CONTACT...
  ```

- S_LST_OF_VAL has one or more referenced records (R).

  ```
  insert into <sometable>...

  select...

  from S_LST_OF_VAL...
  ```

Table 3 lists tables that are modified during upgrades from Release 6.x. Modifications may consist of inserted records (I), updated records (U), or referenced records (R).

Table 3. Tables Modified During Upgrades from Release 6.x

Table...	has records that are inserted (I), updated (U), or referenced (R) during upgrades from Release...			
	6.0.1	6.2.1	6.3	6.3.1
S_ACCELERATOR	IR	IR	IR	IR
S_ACCELRTR_INTL	I	I	I	I
S_ACCNT_ATT	I	I	I	I
S_ACCNT_CHRCTR	R	R	R	R
S_ACCNT_POSTN	IUR	IUR	IUR	IUR
S_ACT_CONTACT	IR	IR	IR	IR
S_ACT_EMP	IUR	IUR	IUR	IUR
S_ACT_ORDER	R	R	R	R
S_ACT_TIMESTAMP	R	R	R	R
S_ACTIVITY_ATT	R	R	R	R
S_ADDR_ORG	IUR	IR	IR	IR
S_ADDR_ORG_INT	UR	R	R	R
S_ADDR_PER	IUR	IR	IR	IR
S_AGREE_ITEM	U	U	U	U
S_AGREE_POSTN	IR	IR	IR	IR
S_APP_VER	U	U	U	U
S_APP_VIEW	R	R	R	R
S_APP_VIEW_RESP	U	U	U	U
S_APPL_WTMPL_IT	U	U	U	U
S_APPLET	R	R	R	R
S_ASGN_ATTR_COL	R	R	R	R
S_ASGN_RULE	U	U	U	U
S_ASSESS	R	R	R	R
S_ASSESS_VAL	U	U	U	U
S_ASSET	UR	UR	UR	UR

Table 3. Tables Modified During Upgrades from Release 6.x

Table...	has records that are inserted (I), updated (U), or referenced (R) during upgrades from Release...			
	6.0.1	6.2.1	6.3	6.3.1
S_ASSET_ACCNT	IR	IR	IR	IR
S_ASSET_WRNTY	R	R	R	R
S_AUC_ALRT_CAT	R	R	R	R
S_AUC_ALRT_DEF	U	U	U	U
S_AUC_ITEM	U	U	U	U
S_BC_ENTRY_TRGT	UR	UR	UR	UR
S_BC_VIEWMODE	I	I	I	I
S_BCSTMSG_DIVN	R	R	R	R
S_BRDCST_MSG	U	U	U	U
S_BU	IUR	IUR	IUR	IUR
S_BUSCOMP	UR	UR	UR	UR
S_C_LST_CHRCTR	I	I	I	I
S_C_LST_COL_MAP	R	R	R	R
S_C_LST_MKT_SEG	R	R	R	R
S_C_LST_POSTN	R	R		
S_C_LST_QRY_CAT			R	R
S_CALL_LST	UR	UR	UR	UR
S_CALL_LST_BU	I	I	I	I
S_CALL_LST_TMPL	I	I	I	I
S_CAMP_CON	UR	UR	UR	UR
S_CERT_ATTR_PER	R	R	R	R
S_CFG_CLASS	IR	IR	IR	IR
S_CFG_ITEM	R	R	R	R
S_CFG_PROP_DEF	IUR	IUR	IUR	IUR
S_CFG_RULE_DEF	I	I	I	I
S_CFG_RULE_SPEC	R	R	R	R

Table 3. Tables Modified During Upgrades from Release 6.x

Table...	has records that are inserted (I), updated (U), or referenced (R) during upgrades from Release...			
	6.0.1	6.2.1	6.3	6.3.1
S_CFG_SCRIPT	R	R	R	R
S_CFG_SCRPT_DEF	IR	IR	IR	IR
S_CFG_VAR_DEF	IUR	IUR	IUR	IUR
S_CFGA_VARDEF	R	R	R	R
S_CHRCTR	IU	IU	IU	IU
S_CHRCTR_VAL	I	I	I	I
S_CM_CFG_USER	IR	IR	IR	IR
S_CM_TELESET	IR	IR	IR	IR
S_CM_USER_QUE	IR	IR	IR	IR
S_CO_BUS_AREA	U	U	U	U
S_COLUMN	UR	UR	UR	UR
S_COMM_REQ			U	U
S_COMM_TMPL	R	R	R	R
S_COMMAND	R	R	R	R
S_COMMUNICATION	U	U	U	U
S_CON_ADDR	R	R	R	R
S_CON_CHRCTR	IR	IR	IR	IR
S_CON_MKT_SEG	R	R	R	R
S_CONTACT	IUR	IUR	IUR	IUR
S_CONTACT_ATT	I	I	I	I
S_CONTACT_BU	I	I	I	I
S_CONTACT_INFO	R	R	R	R
S_CONTACT_REL	U	U	U	U
S_CONTACT_SS	U	U	U	U
S_CONTACT_T	U	U	U	U
S_CONTACT_X	U	U	U	U

Table 3. Tables Modified During Upgrades from Release 6.x

Table...	has records that are inserted (I), updated (U), or referenced (R) during upgrades from Release...			
	6.0.1	6.2.1	6.3	6.3.1
S_CONTROL	U	U	U	U
S_CRSE	R	R	R	R
S_CRSE_AUD	UR	UR	UR	UR
S_CRSE_LIT	UR	UR	UR	UR
S_CRSE_OFFR	UR	UR	UR	UR
S_CRSE_OFFR_TST	IUR	IUR	IUR	IUR
S_CRSE_REG	R	R	R	R
S_CRSE_REL	U	U	U	U
S_CRSE_TST	UR	UR	UR	UR
S_CRSE_TST_LANG	R	R	R	R
S_CRSE_TST_QUES	UR	UR	UR	UR
S_CRSE_TSTRUN	U	U	U	U
S_CS_PATH	UR	UR	UR	UR
S_CS_QUEST	U	U	U	U
S_CS_RUN	U	U	U	U
S_CTI_CFG	R	R	R	R
S_CTI_EXTENSION	R	R	R	R
S_CTI_TELESET	R	R	R	R
S_CTI_USER	R	R	R	R
S_CTLG	UR	UR	UR	UR
S_CTLG_CAT	UR	UR	UR	UR

Table 3. Tables Modified During Upgrades from Release 6.x

Table...	has records that are inserted (I), updated (U), or referenced (R) during upgrades from Release...			
	6.0.1	6.2.1	6.3	6.3.1
S_CTLG_CAT_AUC	R	R	R	R
S_CTLG_CAT_CRSE	R	R	R	R
S_CTLG_CAT_ISS	R	R	R	R
S_CTLG_CAT_LIT	R	R	R	R
S_CTLG_CAT_ORG	R	R	R	R
S_CTLG_CAT_PROD	UR	UR	UR	UR
S_CTLG_CAT_SR	R	R	R	R
S_CTLG_CAT_SRC	R	R	R	R
S_CTLGCT_RESITM	R	R	R	R
S_CURRCLM	R	R	R	R
S_CURRCLM_CRSE	U	U	U	U
S_CURRCLM_PER	U	U	U	U
S_CURRCLM_STEP	I	I	I	I
S_CUST_SURVEY	U	U	U	U
S_DD_HIERATR			U	U
S_DD_HIERATRVAL			U	U
S_DD_MEAS_ATTR			IR	IR
S_DD_PROGLSTFMT	R	R		
S_DEAL_DCP	R	R	R	R
S_DMND_CRTN_PRG	IUR	IUR	IUR	IUR
S_DNB_CON_MRC	R	R	R	R
S_DNB_CONTACT	R	R	R	R
S_DNB_ORG	U	U	U	U

Table 3. Tables Modified During Upgrades from Release 6.x

Table...	has records that are inserted (I), updated (U), or referenced (R) during upgrades from Release...			
	6.0.1	6.2.1	6.3	6.3.1
S_DOC_AGREE	UR	UR	UR	UR
S_DOC_AGREE_BU	I	I	I	I
S_DOC_ORDER	U	U	U	U
S_DOC_PPSL	R	R	R	R
S_DOC_PPSL_BU	I	I	I	I
S_DOC_QUOTE	UR	UR	UR	UR
S_DOC_QUOTE_BU	I	I	I	I
S_DOCK_TXN_LOG	R	R	R	R
S_DOCK_VIS_RULE	U	U	U	U
S_DUAL	R	R	R	R
S_EIM_FK_MAP	UR	UR	UR	UR
S_EIM_FK_MAPCOL	UR	UR	UR	UR
S_EMP_INDUST	R	R	R	R
S_EMP_PER	IUR	IUR	IUR	IUR
S_EMP_PERF_MEAS	U	U	U	U
S_EMPLOYEE_ATT	R	R	R	R
S_EMPLOYEE_X	R	R	R	R
S_EMPLOYEE_XM	R	R	R	R
S_ESCL_REQ	UR	R	R	R
S_ESCL_RULE	R			
S_EVT_ACT	UR	UR	UR	UR
S_EVT_ACT_X	R	R	R	R
S_EVT_FUL_REQ	U	U	U	U
S_EVT_MAIL	I	I	I	I
S_EVT_STAFF	R	R	R	R

Table 3. Tables Modified During Upgrades from Release 6.x

Table...	has records that are inserted (I), updated (U), or referenced (R) during upgrades from Release...			
	6.0.1	6.2.1	6.3	6.3.1
S_EXP_ITEM	U	U	U	U
S_EXP_ITEM_TYPE	R	R	R	R
S_EXP_RPT	U	U	U	U
S_FIELD	R	R	R	R
S_FUL_REQ_CNTNT	R	R	R	R
S_IC_CALC	UR	U	U	U
S_IC_CALC_IT	UR	U	U	U
S_IC_CALC_RUN	IUR	U	U	U
S_IC_CMPNT_FLTR	I	I	I	I
S_IC_DRAW		R	R	R
S_IC_DRAW_BAL	I	I	I	I
S_IC_DRAW_DIVN		I	I	I
S_IC_DRAW_PER		IU	IU	IU
S_IC_FLTR	U			
S_IC_PAYMENT	UR	UR	UR	UR
S_IC_PERF_HIST	U	U	U	U
S_IC_PLCMP_DIVN		I	I	I
S_IC_PLCMP_PER	IR	IR	IR	IR
S_IC_PLN_CMPNT	UR	UR	UR	UR
S_IC_RATE	IUR	IUR	IUR	IUR
S_IC_TXN	UR	UR	UR	UR
S_IC_TXN_IT	UR	U	U	U
S_IC_TXN_POSTN	U			
S_INDUST	R	R	R	R
S_INT_CKEY	IUR	IUR	IUR	IUR

Table 3. Tables Modified During Upgrades from Release 6.x

Table...	has records that are inserted (I), updated (U), or referenced (R) during upgrades from Release...			
	6.0.1	6.2.1	6.3	6.3.1
S_INT_CKEY_FLD	I	I	I	I
S_INT_COMP	UR	UR	UR	UR
S_INT_COMPMAP	R	R	R	R
S_INT_FIELD	UR	UR	UR	UR
S_INT_FLDMAP	R	R	R	R
S_INT_OBJ	R	R	R	R
S_INT_OBJMAP	R	R	R	R
S_INV_TXN	R	R	R	R
S_INV_TXN_BU	IR	IR	IR	IR
S_INVLOC	UR	UR	UR	UR
S_INVOICE_ITEM	U	U	U	U
S_ISS	R	R	R	R
S_ISS_BU	I	I	I	I
S_LIST_COLUMN	U	U	U	U
S_LIT	UR	UR	UR	UR
S_LIT_PRTL_DISP			I	I
S_LOCALE	R	R	R	R
S_LST_OF_VAL	UR	UR	UR	UR
S_MBL_DVC	I	I	I	I
S_MBL_DVC_ADDR	I	I	I	I
S_MDF	R	R	R	R
S_MDF_ALLOC	R	R	R	R
S_MDF_BU	IR	IR	IR	IR
S_MKT_SEG	UR	UR	UR	UR
S_MKT_SEG_VAL	R	R	R	R

Table 3. Tables Modified During Upgrades from Release 6.x

Table...	has records that are inserted (I), updated (U), or referenced (R) during upgrades from Release...			
	6.0.1	6.2.1	6.3	6.3.1
S_NEWS_TOPIC	I	I	I	I
S_NEWSTPC_USER	I	I	I	I
S_NOTE	R	R	R	R
S_NOTE_DCP	IR	IR	IR	IR
S_NOTE_USR_MSG	I	I	I	I
S_OLAP_CAT	U	U	IUR	IUR
S_OLAP_REPORT			IR	IR
S_OLAP_RPT_CAT			I	I
S_OLAPCAT_POSTN	R	R	IR	IR
S_ONL_OBJ_PARTY	R	R	R	R
S_ONL_OBJECT	U	U	U	U
S_ONL_PROJ_PRTY	R	R	R	R
S_ONL_PROJECT	U	U	U	U
S_ONL_PS_CTGRY	R	R	R	R

Table 3. Tables Modified During Upgrades from Release 6.x

Table...	has records that are inserted (I), updated (U), or referenced (R) during upgrades from Release...			
	6.0.1	6.2.1	6.3	6.3.1
S_OPTY	UR	UR	UR	UR
S_OPTY_BU	I	I	I	I
S_OPTY_CHRCTR	R	R	R	R
S_OPTY_INDUST	R	R	R	R
S_OPTY_ORG	R	R	R	R
S_OPTY_POSTN	UR	UR	UR	UR
S_OPTY_PROD	R	R	R	R
S_OPTY_PROJ	R	R	R	R
S_ORD_CRDT_ASGN	I	I	I	I
S_ORDER	UR	UR	UR	UR
S_ORDER_BU	IU	IU	IU	IU
S_ORDER_DTL	R	R	R	R
S_ORDER_ITEM	UR	UR	UR	UR
S_ORDER_POSTN	IUR	IUR	IUR	IUR
S_ORDER_SHIP	R	R	R	R
S_ORDER_TYPE	R	R	R	R
S_ORG_BU	I	I	I	I
S_ORG_CRDT_AREA	R	R	R	R
S_ORG_EXT	IUR	IUR	IUR	IUR
S_ORG_EXT_SS	U	U	U	U
S_ORG_EXT_T	UR	UR	UR	UR
S_ORG_EXT_X	U	U	U	U
S_ORG_FUL	U	U		
S_ORG_FUL_PER	R	R		
S_ORG_GROUP	UR	UR	UR	UR
S_ORG_GROUP_BU	I	I	I	I
S_ORG_GROUP_X	U	U	U	U
S_ORG_INT_ATT	R	R	R	R

Table 3. Tables Modified During Upgrades from Release 6.x

Table...	has records that are inserted (I), updated (U), or referenced (R) during upgrades from Release...			
	6.0.1	6.2.1	6.3	6.3.1
S_ORG_MKT_SEG	R	R	R	R
S_ORG_PRTNR	U	U	U	U
S_ORG_PRTNR_ATT	R	R	R	R
S_ORG_PRTNR_EXP	R	R	R	R
S_ORG_PRTNR_RGN	R	R	R	R
S_ORG_REL	R	R	R	R
S_ORGGRP_POSTN	UR	UR	UR	UR
S_ORGGRP_TERR	R	R	R	R
S_OU_PRTNR_TIER	IR	IR	IR	IR
S_OU_PRTNR_TYPE	I	I	I	I
S_PARTY	IUR	IUR	IUR	IUR
S_PARTY_PER	IR	IR	IR	IR
S_PARTY_REL	U	U	U	U
S_PARTY_RPT_REL	IR	IR	IR	IR
S_PER_COMM_ADDR	R	R	R	R
S_PER_PRTNRAPPL	U	U	U	U
S_PER_RESP	IR	IR	IR	IR
S_PER_WRK_EXP	R	R	R	R
S_PERIOD	R	R	R	R
S_PICKLIST	U	U	U	U
S_POSTN	UR	UR	UR	UR
S_POSTN_CON	I	I	I	I
S_PPSL_SECT	U	U	U	U
S_PRD_DFCT_EMP	R	R	R	R
S_PRD_DFCT_TAG	R	R	R	R
S_PRDFCT_PRJITM	IR	IR	IR	IR
S_PRDFCT_SYMP	R	R	R	R
S_PREPAY_BAL	U	U	U	U

Table 3. Tables Modified During Upgrades from Release 6.x

Table...	has records that are inserted (I), updated (U), or referenced (R) during upgrades from Release...			
	6.0.1	6.2.1	6.3	6.3.1
S_PREPAYBAL_PTY	R	R	R	R
S_PRI_LST	UR	UR	UR	UR
S_PRI_LST_BU	IR	IR	IR	IR
S_PRI_LST_ITEM	U	U	U	U
S_PRILST_MKTSEG	R	R	R	R
S_PRIMDL_FCTR	U	U	U	U
S_PROD_CFGVER	IR	IR	IR	IR
S_PROD_DEFECT	UR	UR	UR	UR
S_PROD_DFCT_BU	I	I	I	I
S_PROD_INT	IUR	IUR	IUR	IUR
S_PROD_INT_BU	U	U	U	U
S_PROD_INT_CRSE	I	I	I	I
S_PROD_ITEM	IUR	IUR	IUR	IUR
S_PROJ	IUR	IUR	IUR	IUR
S_PROJ_BU	I	I	I	I
S_PROJ_POSTN	I	I	I	I
S_PROJ_RSRC	IR	IR	IR	IR
S_PROJITEM	IUR	IUR	IUR	IUR
S_PROJITEM_ATT	I	I	I	I
S_PROJITEM_DIVN	R	R	R	R
S_PROJITEM_ORG	I	I	I	I
S_PROJITEM_REL	R	R	R	R
S_PROJITEM_RSRC	IR	IR	IR	IR
S_PROJITEM_TAG	R	R	R	R
S_PRSP_CONTACT	IUR	IUR	IUR	IUR
S_PRTL_DISP			IR	IR

Table 3. Tables Modified During Upgrades from Release 6.x

Table...	has records that are inserted (I), updated (U), or referenced (R) during upgrades from Release...			
	6.0.1	6.2.1	6.3	6.3.1
S_PRTNRAPP_REGN	R	R	R	R
S_PRTNRAPPL_EXP	R	R	R	R
S_PRTNRAPPL_IND	R	R	R	R
S_PRTNRAPPL_MKT	R	R	R	R
S_PTSHP_MKT_SEG	R	R	R	R
S_QRY_CAT			R	R
S_QTA_ACHIEVE	U			
S_QTA_INCENTV	UR	UR	UR	UR
S_QTA_OBJ	UR	UR	UR	UR
S_QTA_OBJASGN	R	R	R	R
S_QTA_OBJCRT	U	U	U	U
S_QTA_ORG_INT		R	R	R
S_QTA_PLAN	UR	R	R	R
S_QTA_POSTN	UR	UR	UR	UR
S_QTAOBJAS_DIVN		I	I	I
S_QTAOBJAS_PER	U	IU	IU	IU

Table 3. Tables Modified During Upgrades from Release 6.x

Table...	has records that are inserted (I), updated (U), or referenced (R) during upgrades from Release...			
	6.0.1	6.2.1	6.3	6.3.1
S_QTASGN_POSTN	U	U	U	U
S_QUOTA	R	R	R	R
S_QUOTE_ITEM	U	U	U	U
S_QUOTE_SOLN	R	R	R	R
S_REPOSITORY	UR	UR	UR	UR
S_RESITEM	U	U	U	U
S_RESP	UR	UR	UR	UR
S_REVN	IR	IR	IR	IR
S_RULE_DESIGNER	R	R	R	R
S_RULE_TMPL	U	U	U	U
S_SALES_METHOD	R	R	R	R
S_SALES_MTHD_BU	IR	IR	IR	IR
S_SHIPMENT	U	U	U	U
S_SRC	IUR	IUR	IUR	IUR
S_SRC_BU	I	I	I	I
S_SRC_COST	U	U	U	U
S_SRC_DCP	IR	IR	IR	IR
S_SRC_EVT	IUR	IUR	IUR	IUR
S_SRC_EVT_REG	IUR	IUR	IUR	IUR
S_SRC_INDUST	R	R	R	R
S_SRC_PAYMENT	IUR	IUR	IUR	IUR
S_SRC_POSTN	I	I	I	I
S_SRC_VENUE	R	R	R	R
S_SRM_ACT_PARAM	R	R	R	R
S_SRM_ACTION	R	R	R	R
S_SRM_REQUEST	R	R	R	R
S_SRV_ACT	I	I	I	I

Table 3. Tables Modified During Upgrades from Release 6.x

Table...	has records that are inserted (I), updated (U), or referenced (R) during upgrades from Release...			
	6.0.1	6.2.1	6.3	6.3.1
S_SRV_REQ	UR	UR	UR	UR
S_SRV_REQ_BU	I	I	I	I
S_STG_LIT	R	R	R	R
S_STORE_COND	U	U	U	U
S_SYMPTOM	R	R	R	R
S_SYS_PREF	U	U	U	U
S_TABLE	UR	UR	UR	UR
S_TERR	R			
S_TMPL_PLANITEM	R	R	R	R
S_TMPL_PLNIT_BU	IR	IR	IR	IR
S_TMSHT	U	U	U	U
S_TMSHT_ITEM	UI	UI	UI	UI
S_TST_QUES	I	I	I	I
S_UK_ATTJOIN	R	R	R	R
S_USER	IR	IR	IR	IR
S_USER_KEY	UR	UR	UR	UR
S_USER_KEY_ATT	UR	UR	UR	UR

Table 3. Tables Modified During Upgrades from Release 6.x

Table...	has records that are inserted (I), updated (U), or referenced (R) during upgrades from Release...			
	6.0.1	6.2.1	6.3	6.3.1
S_USR_MSG	IR	IR	IR	IR
S_USR_MSG_ATT	I	I	I	I
S_USR_MSG_CAT	I	I	I	I
S_USR_MSG_PREF	I	I	I	I
S_VIEW	U	U	U	U
S_VIEW_WTMPL_IT	U	U	U	U
S_WEB_PG_ITM	U	U	U	U
S_WEB_TMPL	R	R	R	R
S_WEB_TMPL_FILE	I	I	I	I
S_WF_PROC_FLOW	U	U	U	U
S_WF_STEP	U	U	U	U
S_WF_STEP_ARG	U	U	U	U
S_WRNTY_CVRG	R	R	R	R
S_WRNTY_CVRG_BU	IR	IR	IR	IR
S_XA_ATTR	IR	IR	IR	IR
S_XA_CLASS	IUR	IUR	IUR	IUR
S_XA_CLASS_REL	I	I	I	I

Table 4 lists tables that are modified during upgrades from Release 7.0.x. Modifications may consist of inserted records (I), updated records (U), or referenced records (R).

Table 4. Tables Modified During Upgrades from Release 7.0.x

Table...	has records inserted (I), updated (U), or referenced (R) during upgrades from Release...	
	7.0.3	7.0.4
S_ACCNT_CHRCTR	R	R
S_ACCNT_POSTN	U	U
S_ACT_EMP	UR	UR
S_ACT_ORDER	R	R
S_ACTIVITY_ATT	R	R
S_ADDR_ORG	R	R
S_AGREE_ITEM	U	U
S_AGREE_TERMS	U	U
S_APP_VER	U	U
S_APP_VIEW	R	R
S_APP_VIEW_RESP	U	U
S_ASSESS	R	R
S_ASSESS_VAL	U	U
S_ASSET	UR	UR

Table 4. Tables Modified During Upgrades from Release 7.0.x

Table...	has records inserted (I), updated (U), or referenced (R) during upgrades from Release...	
	7.0.3	7.0.4
S_ASSET_ACCNT	I	I
S_BC_ENTRY_TRGT	UR	UR
S_BCSTMSG_DIVN	R	R
S_BRDCST_MSG	U	U
S_BU	R	R
S_CALL_LST	R	R
S_CALL_LST_BU	I	I
S_CAMP_CON	UR	UR
S_CB_ASSET	IR	IR
S_CB_ASSET_AGRP	I	I
S_CB_ASSET_DATA	I	I
S_CB_ASSET_REF	I	I
S_CB_ASSET_VER	IR	IR

Table 4. Tables Modified During Upgrades from Release 7.0.x

Table...	has records inserted (I), updated (U), or referenced (R) during upgrades from Release...	
	7.0.3	7.0.4
S_CB_CNTNT_SET	IR	IR
S_CB_CNTTYP_EXT	R	R
S_CHRCTR	U	U
S_COLUMN	UR	UR
S_COMMUNICATION	U	U
S_CON_CHRCTR	R	R
S_CONTACT	UR	UR
S_CONTACT_BU	I	I
S_CONTACT_INFO	R	R
S_CONTACT_REL	U	U
S_CONTACT_T	U	U
S_CRSE_JOBRL	UR	UR
S_CRSE_OFFR_TST	UR	UR
S_CRSE_TST	U	U
S_CRSE_TST_LANG	R	R

Table 4. Tables Modified During Upgrades from Release 7.0.x

Table...	has records inserted (I), updated (U), or referenced (R) during upgrades from Release...	
	7.0.3	7.0.4
S_CRSE_TST_QUES	R	R
S_CRSE_TSTRUN	R	R
S_CRSE_TSTRUN_Q	U	U
S_CS_PATH	U	U
S_CS_QUEST	U	U
S_CURRCLM	R	R
S_CURRCLM_CRSE	U	U
S_CURRCLM_JOBRL	UR	UR
S_CURRCLM_STEP	I	I
S_CUST_SURVEY	U	U
S_DD_HIERATR	U	U
S_DD_HIERATRVAL	U	U
S_DEAL_DCP	R	R
S_DMND_CRTN_PRG	U	U
S_DNB_ORG	U	U
S_DOC_AGREE	R	R

Table 4. Tables Modified During Upgrades from Release 7.0.x

Table...	has records inserted (I), updated (U), or referenced (R) during upgrades from Release...	
	7.0.3	7.0.4
S_DOC_ORDER	U	U
S_DOC_QUOTE	UR	UR
S_DOCK_TXN_LOG	R	R
S_DOCK_VIS_RULE	U	U
S_DUAL	R	R
S_EIM_FK_MAP	UR	UR
S_EIM_FK_MAPCOL	UR	UR
S_EMP_INDUST	R	R
S_EMP_PER	U	U
S_EMP_PERF_MEAS	U	U
S_ESCL_REQ	R	R
S_EVT_ACT	UR	UR
S_EVT_FUL_REQ	U	U
S_EVTLOC	IUR	IUR

Table 4. Tables Modified During Upgrades from Release 7.0.x

Table...	has records inserted (I), updated (U), or referenced (R) during upgrades from Release...	
	7.0.3	7.0.4
S_EVTLOC_REL	I	I
S_EXP_ITEM	U	U
S_EXP_ITEM_TYPE	R	R
S_EXP_RPT	U	U
S_FCST	R	R
S_FCST_ITEM	U	U
S_FCSTSER_DATE	R	R
S_FUL_REQ_CNTNT	R	R
S_IC_CALC	U	U
S_IC_CALC_IT	U	U
S_IC_CALC_RUN	U	U
S_IC_CMPNT_FLTR	I	I
S_IC_DRAW	R	R
S_IC_DRAW_BAL	I	I

Table 4. Tables Modified During Upgrades from Release 7.0.x

Table...	has records inserted (I), updated (U), or referenced (R) during upgrades from Release...	
	7.0.3	7.0.4
S_IC_DRAW_PER	U	U
S_IC_PAYMENT	UR	UR
S_IC_PERF_HIST	U	U
S_IC_PLCMP_PER	IR	IR
S_IC_PLN_CMPNT	R	R
S_IC_RATE	IUR	IUR
S_IC_TXN	UR	UR
S_IC_TXN_IT	U	U
S_INDUST	R	R
S_INT_COMP	R	R
S_INT_COMPMAP	R	R
S_INT_FLDMAP	R	R
S_INT_OBJ	R	R
S_INT_OBJMAP	R	R

Table 4. Tables Modified During Upgrades from Release 7.0.x

Table...	has records inserted (I), updated (U), or referenced (R) during upgrades from Release...	
	7.0.3	7.0.4
S_INVOICE_ITEM	U	U
S_JOB_PROFILE	IR	IR
S_LIST_COLUMN	U	U
S_LIT	R	R
S_LIT_PRTL_DISP	R	R
S_LOCALE	R	R
S_LST_OF_VAL	UR	UR
S_NEWS_TOPIC	I	I
S_NEWSTPC_USER	I	I
S_ONL_OBJECT	U	U
S_ONL_PS_CTGRY	R	R
S_OPTY	U	U
S_OPTY_CHRCTR	R	R
S_OPTY_CON	U	U

Table 4. Tables Modified During Upgrades from Release 7.0.x

Table...	has records inserted (I), updated (U), or referenced (R) during upgrades from Release...	
	7.0.3	7.0.4
S_OPTY_ORG	R	R
S_ORD_CRDT_ASGN	I	I
S_ORDER	UR	UR
S_ORDER_DTL	R	R
S_ORDER_ITEM	U	U
S_ORDER_POSTN	R	R
S_ORDER_SHIP	R	R
S_ORG_BU	U	U
S_ORG_EXT	UR	UR
S_ORG_EXT_T	U	U
S_ORG_GROUP	UR	UR
S_ORG_GROUP_BU	I	I
S_ORG_PRTNR	U	U
S_ORG_PRTNR_ATT	R	R

Table 4. Tables Modified During Upgrades from Release 7.0.x

Table...	has records inserted (I), updated (U), or referenced (R) during upgrades from Release...	
	7.0.3	7.0.4
S_ORG_PRTNR_RGN	R	R
S_ORG_REL	R	R
S_ORGGRP_POSTN	U	U
S_PARTY	UR	UR
S_PARTY_GROUP	I	I
S_PARTY_GRP_BU	I	I
S_PARTY_PER	R	R
S_PER_COMM_ADDR	R	R
S_PER_WRK_EXP	R	R
S_PERF_RVW	R	R
S_PERF_RVW_CMP	R	R
S_PERF_RVW_COMP	I	I
S_PERF_RVW_TMPL	IUR	IUR
S_PERIOD	R	R

Table 4. Tables Modified During Upgrades from Release 7.0.x

Table...	has records inserted (I), updated (U), or referenced (R) during upgrades from Release...	
	7.0.3	7.0.4
S_POOL_QUES	I	I
S_POSTN	UR	UR
S_PPSL_SECT	U	U
S_PRD_DFCT_EMP	R	R
S_PRD_DFCT_TAG	R	R
S_PRI_LST_ITEM	U	U
S_PROD_CFGVER	R	R
S_PROD_DEFECT	U	U
S_PROD_INT	IUR	IUR
S_PROD_INT_CRSE	I	I
S_PROJ	IUR	IUR
S_PROJ_BU	I	I
S_PROJ_RSRC	IR	IR
S_PROJITEM	IUR	IUR

Table 4. Tables Modified During Upgrades from Release 7.0.x

Table...	has records inserted (I), updated (U), or referenced (R) during upgrades from Release...	
	7.0.3	7.0.4
S_PROJITEM_ATT	I	I
S_PROJITEM_ORG	I	I
S_PROJITEM_REL	R	R
S_PROJITEM_RSRC	I	I
S_PROJITEM_TAG	R	R
S_PROJRSRC_SKL	U	U
S_PROJRSRC_SKLI	U	U
S_PRSP_CONTACT	U	U
S_PRTL_DISP	R	R
S_PRTL_ITEM	IUR	IUR
S_PRTL_ITM_AGRP	I	I
S_PRTL_PAGE	IUR	IUR
S_PRTL_PG_AGRP	I	I
S_PRTL_SCTN_ITM	IR	IR

Table 4. Tables Modified During Upgrades from Release 7.0.x

Table...	has records inserted (I), updated (U), or referenced (R) during upgrades from Release...	
	7.0.3	7.0.4
S_PRTL_SECTION	IUR	IUR
S_QTA_INCENTV	UR	UR
S_QTA_OBJ	R	R
S_QTA_PLAN	R	R
S_QTA_POSTN	UR	UR
S_QTAOBJAS_PER	U	U
S_QTASGN_POSTN	U	U
S_QUOTE_ITEM	U	U
S_REPOSITORY	UR	UR
S_RESP	R	R
S_REVN	R	R
S_RVW_COMP	IU	IU
S_SCHM_PHASE	R	R
S_SCHM_PHS_USG	R	R

Table 4. Tables Modified During Upgrades from Release 7.0.x

Table...	has records inserted (I), updated (U), or referenced (R) during upgrades from Release...	
	7.0.3	7.0.4
S_SCHM_PROC	R	R
S_SCHM_PROC_VAR	R	R
S_SCHM_STEP	R	R
S_SCHM_STEP_DEP	R	R
S_SCHMPROC_DBMS	R	R
S_SCHMST_DBSCPT	R	R
S_SCHMSTEP_SCPT	R	R
S_SCREEN	R	R
S_SHIPMENT	U	U
S_SRC	IUR	IUR
S_SRC_BU	I	I
S_SRC_COST	U	U
S_SRC_DCP	R	R
S_SRC_EVT	IUR	IUR

Table 4. Tables Modified During Upgrades from Release 7.0.x

Table...	has records inserted (I), updated (U), or referenced (R) during upgrades from Release...	
	7.0.3	7.0.4
S_SRC_EVT_REG	U	U
S_SRC_ORG	R	R
S_SRC_PAYMENT	IUR	IUR
S_SRC_VENUE	IR	IR
S_SRCH_CTRESFLD	U	U
S_SRCH_INDX_FLD	U	U
S_SRM_ACT_PARAM	R	R
S_SRM_ACTION	R	R
S_SRV_ACT	I	I
S_SRV_REQ	R	R
S_SRV_REQ_BU	I	I
S_SYS_PREF	U	U
S_TABLE	UR	UR
S_TMPL_PLANITEM	R	R
S_TMSHT	U	U
S_TMSHT_ITEM	U	U
S_TST_QUES	R	R
S_UK_ATTJOIN	R	R
S_USER	IR	IR
S_USER_KEY	UR	UR
S_USER_KEY_ATT	UR	UR

Table 4. Tables Modified During Upgrades from Release 7.0.x

Table...	has records inserted (I), updated (U), or referenced (R) during upgrades from Release...	
	7.0.3	7.0.4
S_USERLIST	I	I
S_USERLIST_BU	I	I
S_USR_MSG	U	U
S_WEB_CNTNT	R	R
S_WEB_CNTNT_LNK	R	R
S_WF_PROC_FLOW	U	U
S_WF_STEP	U	U
S_XA_CLASS	U	U

Table 5 lists tables that are modified during upgrades from Release 7.5.1 to Release 7.5.2. Modifications may consist of updated records (U), or referenced records (R).

NOTE: Upgrades from Release 7.5.1 to Release 7.5.2 are described in a Siebel Applications Technical Note on SupportWeb. See *Technical Note 383: Upgrade from Siebel version 7.5.1 to Siebel version 7.5.2*.

Table 5. Tables Modified During Upgrades from Release 7.5.1

Table...	has records that are updated (U) or referenced (R) during upgrade from Release 7.5.1
S_AGREE_ITEM	U
S_APP_VER	U
S_ASSET	U
S_BU	R
S_COLUMN	U R
S_CONTACT	R
S_CONTACT_REL	U
S_DD_HIERATR	U
S_DD_HIERATRVAL	U
S_DOC_ORDER	U
S_DOCK_TXN_LOG	R
S_DOCK_VIS_RULE	U

Table 5. Tables Modified During Upgrades from Release 7.5.1

Table...	has records that are updated (U) or referenced (R) during upgrade from Release 7.5.1
S_DUAL	R
S_EIM_FK_MAP	U R
S_EIM_FK_MAPCOL	U R
S_ESCL_REQ	R
S_INT_COMP	R
S_INT_COMPMAP	R
S_INT_FLDMAP	R
S_INT_OBJ	R
S_INT_OBJMAP	R
S_LIST_COLUMN	U
S_LST_OF_VAL	U R
S_NEWS_TOPIC	U
S_ORDER	R
S_ORDER_ITEM	U
S_ORG_EXT	R
S_ORG_EXT_T	U
S_ORG_GROUP	R
S_ORG_GROUP_BU	U
S_ORGGRP_POSTN	R
S_PARTY	R
S_POSTN	R
S_QUOTE_ITEM	U
S_REPOSITORY	U R
S_SCHM_PHASE	R
S_SCHM_PHS_USG	R
S_SCHM_PROC	R

Table 5. Tables Modified During Upgrades from Release 7.5.1

Table...	has records that are updated (U) or referenced (R) during upgrade from Release 7.5.1
S_SCHM_PROC_VAR	R
S_SCHM_STEP	R
S_SCHM_STEP_DEP	R
S_SCHMPROC_DBMS	R
S_SCHMST_DBSCPT	R
S_SCHMSTEP_SCPT	R
S_SRM_ACT_PARAM	R
S_SRM_ACTION	R
S_SYS_PREF	U
S_TABLE	U R
S_UK_ATTJOIN	R
S_USER	R
S_USER_KEY	U R
S_USER_KEY_ATT	U R
S_WF_PROC_FLOW	U
S_WF_STEP	U

Tables Seeded During Upgrade to Release 7.5.2

Table 6 lists seeded tables for upgrades to Release 7.5.2.

Table 6. Tables Seeded During Upgrades to Release 7.5.2

Table
S_ACTION_ARG
S_ACTION_DEFN
S_APP_QUERY
S_APP_VIEW
S_APP_VIEW_RESP
S_ASGN_GRP
S_ASGN_GRP_EMP
S_ASGN_GRP_OBJ
S_ASGN_GRP_POSTN
S_ASGN_RULE
S_ASGN_RULE_ITEM
S_ASGN_WL_OBJ
S_ASGN_WL_OBJ_COL
S_AUDIT_BUSCOMP
S_AUDIT_FIELD
S_BC_ENTRY_TRGT
S_BRWSR
S_BRWSR_CPBLTY
S_BRWSR_ELM_EVT
S_BRWSR_FEATURE

Table 6. Tables Seeded During Upgrades to Release 7.5.2

Table
S_BU
S_BUSCOMP_DMAP
S_BUSOBJ_DMAP
S_CB_ASSET
S_CB_ASSET_AGRP
S_CB_ASSET_DATA
S_CB_ASSET_VER
S_CB_CNTNT_TYPE
S_CB_CNTTYP_EXT
S_CM_CFG_PROF
S_CM_CMD
S_CM_CMD_DATA
S_CM_CMD_PARM
S_CM_CMD_REL
S_CM_CNCTR
S_CM_CNCTR_PARM
S_CM_DATA_PARM
S_CM_EVTHDLR
S_CM_EVTLOG
S_CM_EVTRESP
S_CM_HDLR_PARM
S_CM_LOG_PARM

Table 6. Tables Seeded During Upgrades to Release 7.5.2

Table
S_CM_MEDIA
S_CM_MEDIA_PROP
S_CM_PROF
S_CM_PROF_PARM
S_CM_RESP_LOG
S_CM_RESP_PARM
S_COLLAB_CMD
S_CONTACT
S_CONTACT_BU
S_CONTXT_BC
S_CONTXT_BC_VAR
S_CO_BUS_AREA
S_CS_ANSWR
S_CS_EDGE
S_CS_PAGE
S_CS_PATH
S_CS_PATH_BU
S_CS_PATH_SCPT
S_CS_PATH_SRC
S_CS_QUEST
S_CS_QUEST_SCPT
S_CS_STYLE

Table 6. Tables Seeded During Upgrades to Release 7.5.2

Table
S_CTI_CFG
S_CTI_CFG_PARAM
S_CTI_EXTENSION
S_CTI_USER
S_CTLG
S_CTLG_BU
S_CTLG_CAT
S_CTLG_CAT_PROD
S_CT_ACTION
S_CT_ACTION_SET
S_CT_APLT_RLST
S_CT_APPLET
S_CT_EVENT
S_CT_EVENT_DEF
S_CT_RULE
S_CT_RULE_SET
S_CURCY
S_DB_SCH_VER
S_DCP_PKG_ITEM
S_DCP_RESP_TYPE
S_DOCK_ROUTE
S_DOCK_ROUTE_VR

Table 6. Tables Seeded During Upgrades to Release 7.5.2

Table
S_DOC_CORR
S_DOC_PPSL
S_DOC_PPSL_BU
S_DUAL
S_DYNUI_APPLET
S_DYNUI_VIEW
S_EAI_ADAPTER
S_EAI_ADPT_FLOW
S_EAI_APPLVER
S_EAI_APPL_ADPT
S_EAI_DTEOBJ2MP
S_EAI_FLOW
S_EAI_LOOKUPMAP
S_EAI_QUEUE
S_EMP_PER
S_ESCL_ACTION
S_ESCL_COND
S_ESCL_GROUP
S_ESCL_RULE
S_ETL_CTRYREGN
S_ETL_DATASRC
S_ETL_DAY

Table 6. Tables Seeded During Upgrades to Release 7.5.2

Table
S_ETL_ERRHLP
S_ETL_INDUS_VER
S_ETL_SRC_TABLE
S_ETL_STRINGS
S_ETL_TIME_DAY
S_EVT_ACT
S_EXCH_RATE
S_EXP_ITEM_TYPE
S_FCST_SCENARIO
S_FIELD_DMAP
S_INDUST
S_INT_CHANNEL
S_INT_COMPMAP
S_INT_FLDMAP
S_INT_GROUP
S_INT_INSTANCE
S_INT_MAP_ARG
S_INT_OBJMAP
S_INVLOC
S_INVLOC_BU
S_INVLOC_TYPE

Table 6. Tables Seeded During Upgrades to Release 7.5.2

Table
S_INV_TXN_TYPE
S_LANG
S_LIT
S_LIT_BU
S_LOCALE
S_LOV_REL
S_LST_OF_VAL
S_MPP_FLD_MAP
S_MPP_MAP
S_MPP_OBJ_MAP
S_MSG
S_NEWS_PKG
S_NEWS_TOPIC
S_NODE
S_OLAP_CAT
S_OLAP_REPORT
S_OLAP_RPT_CAT
S_ONL_CATEGORY
S_ORDER_TYPE
S_ORDTYP_MVTTYP
S_ORG_EXT
S_PAGER_COMPANY

Table 6. Tables Seeded During Upgrades to Release 7.5.2

Table
S_PARTMVMT_TYPE
S_PARTY
S_PARTY_GROUP
S_PARTY_GRP_BU
S_PARTY_PER
S_PARTY_RPT_REL
S_PERF_RTNG_SCL
S_PERIOD
S_PERIOD_BU
S_PERIOD_REL
S_PER_RESP
S_POSTN
S_POSTN_CON
S_PPSL_SECT
S_PPSL_SECT_FLD
S_PPSL_SECT_LIT
S_PROC_DEFN
S_PROD_INT
S_PROD_INT_BU
S_PROD_INV_CAT
S_PRTL_ITEM
S_PRTL_PAGE

Table 6. Tables Seeded During Upgrades to Release 7.5.2

Table
S_PRTL_SCTN_ITM
S_PRTL_SECTION
S_RESP
S_RESP_BU
S_RULE_DATATYPE
S_RULE_DESIGNER
S_RULE_TMPL
S_RULE_TMPL_ARG
S_RULE_TMPL_CAT
S_SALES_METHOD
S_SALES_MTHD_BU
S_SA_APPL
S_SA_APPL_CNTNT
S_SA_APPL_FLD
S_SA_CNTNT_TYPE
S_SA_CTLG_NMVAL
S_SA_NMVAL_APPL
S_SA_NMVAL_DEF
S_SGML_DOC_TYPE
S_SM_STATE
S_SM_TRANSITION
S_SRCH_ADM

Table 6. Tables Seeded During Upgrades to Release 7.5.2

Table
S_SRCH_FLD
S_STATE_MODEL
S_STG
S_SVCDSP_RL_PRP
S_SVCDSP_RL_SET
S_SVCDSP_RULE
S_SVC_METRIC
S_SYS_PREF
S_TIMEZONE
S_TMPL_PLANITEM
S_UI_CTRL_STYLE
S_UPG_COMP
S_UPG_COMP_REL
S_UPG_KIT_ITARG
S_UQ_CFG
S_UQ_CFG_PARAM
S_UQ_ROUTE
S_UQ_ROUTE_ESCL
S_UQ_ROUTE_PROP
S_UQ_RTESCL_RUL
S_USER
S_USERLIST

Table 6. Tables Seeded During Upgrades to Release 7.5.2

Table
S_USERLIST_ATT
S_USERLIST_BU
S_USER_PROF
S_USER_ROLE_TSK
S_USER_TASK
S_VARSUB_CONTXT
S_WEB_APPL
S_WEB_ELMNT_EVT
S_WEB_ML_ELMNT
S_WEB_UI_EVENT
S_WF_COND_CRIT

Table 6. Tables Seeded During Upgrades to Release 7.5.2

Table
S_WF_COND_VAL
S_WF_PROC_FLOW
S_WF_PROC_PROP
S_WF_STEP
S_WF_STEP_ARG
S_WF_STEP_BRNCH
S_WI_CTNT_FXUP
S_WI_HTTP_HOST
S_WI_SYMURL
S_WI_SYMURL_ARG
S_WS_BNDNG_DTL
S_WS_OPERATION
S_WS_PORT
S_WS_PORT_TYPE
S_WS_WEBSERVICE
S_XML_NAMESPACE
S_ZD_ENBLD_OBJ
S_ZIPCODE

Preupgrade Tasks 3

This chapter describes the procedures that you need to perform before an upgrade of your development environment or your production environment. To better understand which phase of the upgrade process you are in and what tasks still remain, refer to Figure 5. Figure 5 illustrates where you are in the upgrade process depending on whether you are performing a development or production environment upgrade.

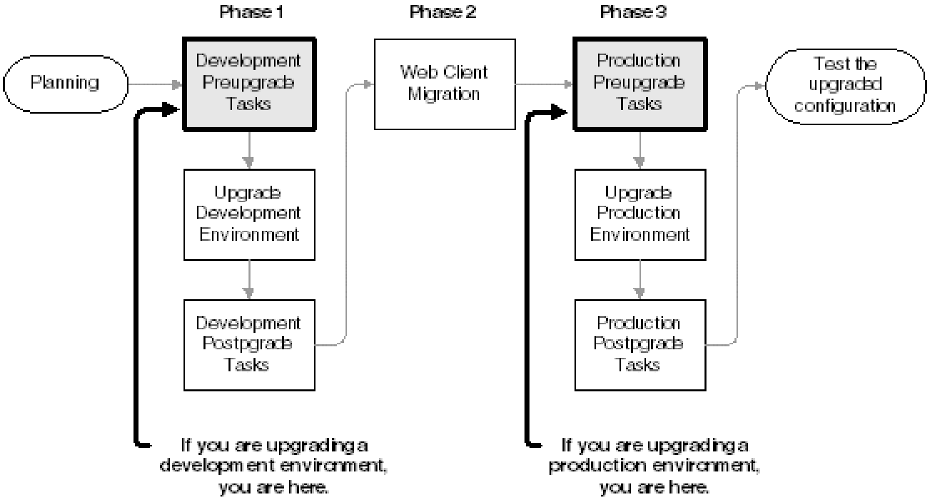

Phase 1
Phase 2
Phase 3
Planning
Development Preupgrade Tasks
Web Client Migration
Production Preupgrade Tasks
Test the upgraded configuration
Upgrade Development Environment
Upgrade Production Environment
Development Postpgrade Tasks
Production Postpgrade Tasks
If you are upgrading a development environment, you are here.
If you are upgrading a production environment, you are here.

Figure 5. You Are Here

If a procedure is specific to one environment, this will be indicated by a caution at the beginning of the procedure; for example:

CAUTION: This procedure only applies to development environment upgrades.

A checklist of preupgrade tasks is provided in Table 7 on page 90. Perform the procedures for the environment that you are upgrading.

- **Development environment upgrades**. Conduct preupgrade procedures for development environments. Skip the procedures that only apply to production environments.
- **Test environment upgrades.** If your deployment has no customizations or configuration changes to Siebel eBusiness Applications, and therefore does not have a development environment, conduct preupgrade and upgrade procedures on a test environment (using a copy of your production database) before you upgrade your production environment. See Appendix B, "Upgrading Without a Development Environment."
- **Production environment upgrades**. Conduct preupgrade procedures for production environments. Skip procedures that only apply to development environments.

Before You Begin

If you have not already done so, copy the Upgrade Planning Worksheet, located in Appendix A, “Upgrade Planning Worksheet,” and fill out the appropriate fields with the following:

- **Siebel Administrator User Name and Password.** This must be a valid RDBMS user name and password, and it must be set up as a Siebel employee. The employee record must have the “Siebel Administrator” responsibility. `SADMIN` is the default administrator user name and password. If this user does not already exist in your database, or does not have “Siebel Administrator” privileges, then you must add this to your database prior to proceeding with the upgrade.
- **Database Tableowner Account User Name and Password.** For DB2 UDB and Oracle, `SIEBEL` is the default tableowner account user name and password for Siebel eBusiness Applications.
- **Siebel Data Tablespace.** Only applicable to Oracle, this is the name of the tablespace on the Oracle server where the Siebel data tables are stored.
- **Siebel Index Tablespace.** The name of the tablespace on the DB2 or Oracle server where the Siebel indexes are stored.
- **Siebel 4-KB Data Tablespace.** The name of the tablespace on the DB2 server where the 4-KB Siebel data tables are stored.
- **Siebel 16-KB Data Tablespace.** The name of the tablespace on the DB2 server where tables reside whose row length equals greater than 4005 bytes, but less than 16,384 bytes.
- **Siebel 32-KB Data Tablespace.** The name of the tablespace on the DB2 server where tables reside whose row length equals up to 32,768 bytes.

NOTE: Siebel tablespaces for the DB2 UDB platform should be database managed tablespaces (DMS) rather than system managed tablespaces (SMS).

A checklist of preupgrade tasks is provided in Table 7.

Table 7. Preupgrade Tasks

		Environment	
	Preupgrade Task	Dev.	Prod.
1	Fill out your copy of Appendix A, “Upgrade Planning Worksheet” with information specific to your RDBMS platform.	✔	✔
2	Prepare to import the new customer repository. See “Preparing to Import the New Customer Repository” on page 91.		✔
3	Verify database sort order. See “Verifying Database Sort Order” on page 91.	✔	✔
4	Save all pending changes. See “Saving All Pending Changes” on page 94.	✔	
5	Back up the database. See “Backing Up the Development or Production Environment Database” on page 95.	✔	✔
6	Save interface table data. See “Saving Interface Table Data” on page 95.	✔	✔
7	Upgrade your RDBMS software. See “Upgrading Your RDBMS Software” on page 98.	✔	✔
8	Verify database server configuration for your RDBMS. See “Verifying Database Server Configuration” on page 99.	✔	✔
9	Preserve custom indexes on tables. See “Preserving Custom Indexes on Tables” on page 119.	✔	✔
10	Disable customized triggers. See “Disabling Customized Triggers” on page 120.	✔	✔
11	Drop customized views. See “Dropping Customized Views” on page 120.	✔	✔
12	Rename the existing development repository from “Siebel Repository” to “Prior Customer Repository.” See “Renaming Your Existing Development Repository” on page 124.	✔	✔
13	Record dock objects and visibility rules. See “Recording Dock Objects and Visibility Rules” on page 126.	✔	✔
14	Prepare your application data. See “Preparing Application Data for Upgrade” on page 127.	✔	✔

Preparing a Database for Upgrade

Before you upgrade your development or production environment, make sure that the database configuration meets the database requirements outlined in "Verifying Database Server Configuration" on page 99, and meets the requirements depicted in *Siebel Server Installation Guide for UNIX*.

Preparing to Import the New Customer Repository

CAUTION: This procedure only applies to production environment upgrades.

Navigate to your development environment's *DBSRVR_ROOT*\PLATFORM directory. Copy the `custrep.dat` file created by your development environment upgrade. Place the `custrep.dat` file in the *DBSRVR_ROOT\PLATFORM* directory for your production environment. This file will be used by the Siebel Upgrade Wizard to import the New Customer Repository.

Verifying Database Sort Order

Sort order (also called collation sequence) is specified during the initial installation of a database and defines the way in which the database will sort character data. Sort order support depends on both the code page of the database and whether it will be used in a development or a production environment.

- **Development environments.** For development databases, you must use a binary sort order due to functional limitations of databases that use a non-binary sort order. See "Sort Order Considerations for Siebel Databases."

- **Production environments.** For production databases, it is strongly recommended that you use binary sort order to prevent possible performance degradation. See "Sort Order Considerations for Siebel Databases."

NOTE: The settings for binary are unique for each database platform; for example, binary sort order is Identity sort order on IBM DB2 UDB. See *Siebel System Requirements and Supported Platforms* to determine supported settings for your database platform.

Sort Order Considerations for Siebel Databases

If your deployment requires that you use a non-binary sort order (for example, if your local language does not use binary sort order), you need to consider several functional limitations that particularly affect development environment upgrades. If these limitations are unacceptable, you should consider recreating your database to use binary sort order.

- You cannot use Siebel Tools to generate a Siebel repository file (SRF) on a database that uses a non-binary sort order.
- You cannot perform a repository merge on a database that uses a non-binary sort order.
- Databases that use non-binary sort order might perform slower than databases that use binary sort order.

See *Siebel System Requirements and Supported Platforms* to verify sort order and code page requirements for your deployment.

Converting Oracle Databases to a Supported Code Page

Several code pages are no longer supported in Release 7.5 because these code pages might contain 1252 characters (for example, the Euro symbol, smart quotes, or hyphens) that will be corrupted during an upgrade. Carefully review *Siebel System Requirements and Supported Platforms* to determine which code pages are supported.

If you want to continue to use a non-Unicode Western European character set code page, you need to covert your Oracle database to a supported code page prior to upgrading to Release 7.5.

CAUTION: Limitations to code page support apply to both the database server and the Client. Check Oracle Client settings for all of your Clients and servers.

To convert your Oracle database to a supported code page, review *Siebel System Requirements and Supported Platforms* for supported database code page information, then refer to Siebel SupportWeb for detailed instructions for using database vendor utilities to convert your Oracle database to a supported code page.

Verifying Sort Order on Oracle Client

See *Siebel System Requirements and Supported Platforms* to verify sort order and code page requirements for your deployment. Siebel Systems strongly recommends binary sort order on your Oracle database, due to limitations of databases that use non-binary sort. (See "Sort Order Considerations for Siebel Databases.")

Sort order on Oracle is determined by the NLS_SORT parameter on the Oracle Client. You must set NLS_SORT to BINARY, or you must choose a NLS_LANG setting that includes binary.

CAUTION: The NLS_LANG parameter must be set to the same value throughout your enterprise, and it must match the database character set. The NLS_LANG parameter is required for conversion from a non-Unicode code page to Unicode; an incorrect setting could lead to data loss.

To verify that your database was created using binary sort order

1 Use SQLPlus to connect the Oracle database.

2 Issue the following query:

```
SQL> SELECT * FROM NLS_DATABASE_PARAMETERS;
```

3 Review the returned parameters for NLS_SORT, and verify that the value for this parameter is BINARY.

- If NLS_SORT has a value of BINARY, then the default sort order is binary and no action is required.
- If NLS_SORT is anything other than BINARY, then you need to recreate the database so that it uses binary sort order. Review "Sort Order Considerations for Siebel Databases," then see *Siebel System Requirements and Supported Platforms* for supported values.

Verifying Sort Order on IBM DB2 UDB

See *Siebel System Requirements and Supported Platforms* to verify sort order and code page requirements for your deployment. Siebel Systems requires Identity sort order on DB2 UDB development databases. Identity sort order is also highly recommended for DB2 UDB production environment databases.

To verify that your database was created using Identity sort order

1. Run the following query on Siebel database:

   ```
   select count (*) from S_APP_VER where '$' > '/'
   ```

2. Review the result.

   - If sort order is correct, the result will be

     ```
     1
     --------------
     0
     (1) record selected.
     ```

   - If sort order is incorrect, you need to recreate the database, using the option:

     ```
     COLLATE USING IDENTITY
     ```

NOTE: Sort order is specified during creation of the database. If you find that your IBM DB2 UDB database was not created using Identity sort order, you must recreate your database using the option `COLLATE USING IDENTITY`.

If sort order is correct, but you are still encountering errors, contact Siebel Technical Services for further analysis.

Saving All Pending Changes

CAUTION: This procedure only applies to development environment upgrades.

Make sure that all project locks in your current Siebel repository have been released to prevent inadvertent loss of development work during the upgrade.

If you are using Siebel Workflow Manager, you must run the Workflow Monitor Agent and Workflow Action Agent to completion before upgrading to Release 7.x. The `S_ESCL_REQ` table should not have any rows if Workflow Manager has completed successfully.

CAUTION: Before starting the upgrade, all locked projects must be checked in and unlocked. If projects are not checked in and unlocked, errors may occur during the repository merge process.

Backing Up the Development or Production Environment Database

Perform a full backup of the development or production database (as appropriate to the environment that you are upgrading), using the utilities specific to your database platform. This backup protects your repositories and environment.

It is strongly recommended that you back up your database repository at key stages of the upgrade:

- Before any upgrade activity is started
- After upgrading the Siebel Database Schema
- After the repository merge (for development environment upgrades)
- After upgrading the Custom Database Schema

Saving Interface Table Data

During the upgrade process, your interface tables will be dropped and then recreated. To retain data in your interface tables, use the appropriate tools for your RDBMS to export data before the upgrade and then import the data after you have completed the upgrade.

NOTE: During the upgrade, all custom indexes on interface tables are dropped from both logical and physical schema.

Securing AIX Memory Allocation Segment Space

Before you run an upgrade on AIX, set the following environment variable on the AIX machine that you are using for the upgrade:

```
setenv LDR_CNTRL LOADPUBLIC@MAXDATA=0x60000000
```

This will prevent a shortage of memory allocation segment space that might occur if the Database Server and Siebel Server are installed on the same machine.

After a successful upgrade, reset this parameter to the original value.

Identifying and Resolving Duplicates

NOTE: Do not perform this procedure for upgrades from Release 7.0.x to Release 7.5 (for example, from Release 7.0.3 or 7.0.4 to Release 7.5), because you completed this during your upgrade to Release 7.0.x.

Before you upgrade to Release 7.5, you need to resolve any duplicates in your Siebel database. To identify duplicates, run the script, `Find_DUP_S_PARTY_ROW_IDs.sql`, located in the *`DBSRVR_ROOT\DatabasePlatform\upgrade`* directory.

This SQL script will generate a list of duplicates (if any) that you must resolve prior to beginning the upgrade to Siebel 7.5.

Preparing Mobile and Dedicated Users for the Upgrade

Complete the following actions before beginning the upgrade of either a development environment or a production environment:

1. Perform a partial synchronization for mobile users, sending all transactions to the server database.

   CAUTION: Mobile users must make no further changes to their local databases until the upgrade has been completed. Any changes made before the upgrade has been completed will be lost when they are reinitialized following the upgrade.

2. Verify that Mobile Clients have synchronized and that all changes have been merged into the server database:

   a. Check that no transaction files remain in the synchronization inbox and outbox for any mobile user. The synchronization inbox for each user is on the Siebel Server.

      UNIX inboxes will be in `/SIEBEL_ROOT/docking/mobileusername.`

      Transaction files are in the format *number.dx*; for example, `00000023.dx`.

   b. Log onto a Siebel eBusiness Application, such as Call Center, as the Siebel Administrator. Use the Server Administration - Server Tasks screen to make sure that each Transaction Merger task has successfully completed.

   c. Verify that Workflow Monitor and Workflow Action agents have processed all pending requests. If Workflow Manager has completed successfully, the `S_ESCL_REQ` table should not have any rows.

3. To prevent synchronization of Mobile Clients with the database server, stop or disable all Siebel Remote components on all Siebel Servers, as described in *Siebel Remote and Replication Manager Administration Guide* and in *Siebel Server Administration Guide*.

4. Disconnect all Web Clients from the Siebel Server by stopping the appropriate Application Object Managers, as described in *Siebel Server Administration Guide*.

5. Make sure that Dedicated Web Clients have disconnected from the Siebel Database Server.

   The method you use to do this will depend on your database. For example, with an Oracle RDBMS, you would stop the primary listener. However, all RDBMS types require starting the database in restricted mode. Refer to the documentation that you received from your RDBMS vendor for more information.

Upgrading Your RDBMS Software

New releases of Siebel eBusiness Applications might require you to upgrade your RDBMS server software. Review the information in *Siebel System Requirements and Supported Platforms* to determine if a database server upgrade is required. Refer to the documentation provided by your database vendor for specific instructions on performing the database upgrade.

CAUTION: If your system does not meet the requirements specified in *Siebel System Requirements and Supported Platforms*, your upgrade will fail.

- If you previously installed DB2 UDB, you must upgrade to the latest DB2 UDB version before upgrading to Release 7.x. Verify that you have updated the database to the current fixpack level as described in *Siebel System Requirements and Supported Platforms*.
- If you are performing an upgrade for DB2 UDB, you must use a DB2 database with 4-KB, 16-KB and 32-KB tablespaces defined on it. Otherwise, your upgrade will not complete successfully.
- The `NextSeq` user defined function (UDF) from Release 6.x was replaced by an alternate version for 7.x, so your DBA needs to verify that only the current version of the UDF is installed. The new function definition accepts only one parameter, and the old one is dropped.

Carefully read the relevant chapters of *Siebel Server Installation Guide* for the RDBMS software you plan to configure and install, and for instructions for installing `NextSeq`.

Make sure that the RDBMS upgrade was successful before you proceed with the upgrade of your Siebel database.

NOTE: Be sure to upgrade your client database connectivity software on all upgraded machines to the version specified in *Siebel System Requirements and Supported Platforms*.

Verifying Database Server Configuration

Before you begin your upgrade, you must verify your database configuration. Later steps of the upgrade process are database-intensive and demand that your database have sufficient resources available. While specific requirements vary by database platform, the consequences of exceeding available resources are the same across platforms: a halted upgrade that requires you to allocate time to adjust the environment and then resume the upgrade.

NOTE: If you want to change your database platform in conjunction with your upgrade, please contact Siebel Technical Services for assistance.

When you upgrade from earlier versions of Siebel eBusiness Applications to Release 7.5, expect database growth in the range of 30%-60% across all platforms. The amount by which it grows depends on a number of factors, such as the size of your database prior to upgrade and the RDBMS software you use.

For estimates of growth percentages for specific RDBMS platforms and releases, see Table 8.

NOTE: For all platforms, the growth percentage will increase depending on how you size your database and configure default storage for database tablespaces. For example, if you set the default storage for your initial or next extent in a given Oracle tablespace to 10 KB, that tablespace will grow by a smaller percentage than if you set it to 100 KB.

Table 8. Siebel Database Expected Growth During a UNIX Upgrade

Release From Which You Are Upgrading	DB2 UDB	Oracle
Release 6.3x	4-K tablespace: 200% 16-K tablespace: 400% 32-K tablespace: reduced by 50%	tables 40%-60% indexes 70%-80%
Release 6.2x	4-K tablespace: 200% 16-K tablespace: 400% 32-K tablespace: reduced by 50%	tables 40%-60% indexes 70%-80%

Table 8. Siebel Database Expected Growth During a UNIX Upgrade

Release From Which You Are Upgrading	DB2 UDB	Oracle
Release 6.0x	4-K tablespace: 200% 16-K tablespace: 400% 32-K tablespace: reduced by 50%	tables 40%-60% indexes 70%-80%
Release 5.6x	4-KB Tablespace: 40% Long Tablespace: 250% 8-KB tablespace to 16-KB tablespace: 350%	40%-80%
Release 5.5x	N/A	40%-80%
Release 5.0x	Same as from Release 5.6x	40%-80%

NOTE: Actual expected growth may vary widely from these estimates, depending on your database configuration, row size of tables, and data content.

Review the configuration requirements for your specific database platform and make sure that your database server configuration meets or exceeds them.

- For server configuration requirements for IBM DB2 UDB, see "DB2 Universal Database Server Configuration" on page 103.
- For server configuration requirements for Oracle, see "Oracle Database Server Configuration" on page 101.

Oracle Database Server Configuration

Before upgrading an Oracle database, verify the following:

- Upgrade your Oracle database and client software to the version referenced in *Siebel System Requirements and Supported Platforms*, using Oracle’s tools and documented procedures.
- Make sure that the system tablespace has sufficient room for growth, since a number of additional objects are created during the upgrade.

Tablespaces

Allow for growth of the database during the upgrade. Make sure that tablespaces have enough free space to accommodate expected growth as estimated in Table 8 on page 99.

pctincrease. For upgrades, you should have a high value `pctincrease` for the tablespaces that contain application tablespaces and indexes.

pctfree. Before you start the upgrade, rebuild some of your larger tables with a large value of `pctfree` for the larger tables (30 or higher). Examples of large tables are:

- `s_contact`
- `s_evt_act`
- `s_srv_req`
- `s_src`
- `s_org_ext`
- `s_addr_org`

NOTE: The reason that you need to increase `pctfree` before the upgrade is that many new columns are added to these tables in Release 7.x. Migrating data into the new columns during the upgrade is likely to cause row chaining, which will cause performance degradation.

Database Parameters

If the database server is a multiple-CPU machine, you can take advantage of parallel index creation. However, if you are executing a Siebel Database Upgrade on an Oracle database server with only one CPU, you must change the Oracle Parallel Index setting in the master upgrade configuration (ucf) file from `Y` to `N`. This step must be performed after Siebel Database Configuration and before execution of the Siebel Upgrade Wizard. If this action is not taken, then certain statements that are executed during the upgrade will fail.

- **Development environment upgrades.** For development environment upgrades, the Oracle Parallel Index parameter (in the master configuration file) defaults to the value `N`.
  - If your database server is a single-CPU machine, the value of this parameter must be set to `N,` or the operation will fail.
  - If your database server has more than one CPU, you can change this parameter to `Y` to yield significance performance benefits.
- **Production environment upgrades.** Siebel eBusiness Applications assume that parallelism is supported for production environment upgrades; therefore, for production environment upgrades, the Oracle Parallel Index parameter (in the master configuration file) defaults to a value of `Y`.

The master configuration file is located in the `SIEBSRVR\bin` directory. The file that you need to modify depends on the Siebel version that you are upgrading from, and whether you are upgrading the Siebel database schema (upgrep) or your custom database schema (upgphys). For example, for upgrades of the Siebel database schema (upgrep) in a development environment from Release 6.0.1 to Release 7.x, you need to edit the master configuration file called `master_upgrep_dev_601.ucf`.

The DBA should set appropriate values according to the Oracle documentation for the following `init.ora` parameters, depending on the number of CPUs on the database server:

- `parallel_max_servers`

  NOTE: The parameter `parallel_max_servers` must be set greater than `1` to enable parallelism.

- `parallel_min_servers`
- `db_block_lru_latches`

The other `init.ora` parameters that the DBA should choose are:

- `sort_area_size`

- `sort_area_retained size`
- `log_buffers`

For more information, see your Oracle documentation.

Rollback Segments

Rollback segments should be appropriately sized to accommodate the largest of transactions. The upgrade may affect some of the largest tables in your implementation of Release 7.x, causing them to grow by as much as 40%.

The shared pool size should be sufficiently large.

DB2 Universal Database Server Configuration

This section provides guidelines for obtaining optimum performance from a DB2 Universal Database.

Before upgrading a DB2 database, verify that your development database server meets or exceeds the following OLTP parameters:

- You have at least 25 primary and 100 secondary logs of at least 32-MB log size.
- Locklist parameters are set to at least 5,000, and preferably 15,000.
- Maxlocks should be set to 20.
- DMS tablespace has at least 25% of free pages. If it does not, you will need to increase tablespace size by adding containers.
- The file system has sufficient space to allow your DMS tablespace to grow.

In addition, verify that the tablespaces are not near their capacity. This can be done by connecting to the database and issuing the following command:

```
db2 list tablespaces show detail
```

Upgrade Instance

After upgrading your RBDMS software (for DB2 UDB, follow the instructions provided by IBM), upgrade your DB2 UDB instance.

To upgrade the instance

- Run the following command on the database server to upgrade to the current version of your RDBMS software:

```
db2updv7 -d DB_NAME
```

Verifying DB2 Permissions on AIX and Solaris

If you are running DB2 UDB on the AIX or Solaris platform, perform the following steps before executing the Siebel database upgrade:

1. Navigate to the instance home directory.
2. Use the following command to verify that the directory `sqllib/function/routine/sqlproc` has write permission for the group:

   ```
   ls -ld sqllib/function/routine/sqlproc
   ```

3. To authorize group write permission, enter the following command:

   ```
   chmod g+w sqllib/function/routine/sqlproc
   ```

Verifying Instance Owner Permissions on DB2 UDB for AIX and Solaris

If you are running DB2 UDB on the AIX or Solaris platforms, verify that the Siebel Database instance owner belongs to the primary group of the fenced user. If the instance owner is not part of this group, you will encounter errors during the Siebel Database upgrade.

Increasing the Number of ODBC Statement Handles

DB2 UDB can quickly run out of ODBC statement handles, depending on the number of business objects your enterprise uses. Because it is difficult to know how many business objects your users actually use, you should increase this number automatically each time you upgrade the DB2 UDB client, or when rebinding database utilities.

Siebel Systems recommends that you increase the number of CLI packages to six by rebinding the CLI packages, using the special DB2 `CLIPKG` bind option.

To rebind the CLI packages

1. Navigate to the subdirectory below, using the appropriate method to your operating system:

   - **UNIX.** From any type of shell, navigate to `sqllib/bnd` in the DB2 instance home directory.
   - **Windows.** On most Windows machines, navigate to `C:\sqllib\bnd` from a DB2 Command window.

2 Connect to the DB2 UDB database.

3 Enter the following command:

```
db2 bind @db2cli.lst blocking all grant public clipkg 6
```

For more information about the DB2 bind command and the `CLIPKG` option, refer to IBM DB2 documentation.

Upgrade-Specific Parameters

The default settings of the parameters in this section should be adjusted for upgrading to Release 7.5. The values recommended in the following pages are guidelines only, and your environment may require adjustments to these values.

After your upgrade has been completed, and prior to running Release 7.5 in a production environment, you must adjust the DB2 parameters described in this section to those values recommended in *Siebel Server Installation Guide* for the operating system you are using. Otherwise, your Siebel Database Server may not provide optimal performance.

See your IBM DB2 technical documentation for additional information on DB2 parameters.

DB2 Database Manager Configuration Parameters

You can set the database configuration parameters using the `update database manager configuration` command of the DB2 Command Line Processor or using the DB2 Control Center.

NOTE: Refer to your IBM DB2 UDB documentation for more information on modifying the database configuration parameters.

Table 9 provides guidelines for setting DB2 Database Manager configuration parameters specifically for an optimal upgrade of your Siebel Database. Set these parameters for each DB2 instance. Use the configuration information below for the listed parameters. For parameters not listed in this table, accept the default settings.

Table 9. DB2 Database Manager Configuration Parameters

Parameter	Explanation	Setting
SHEAPTHRES	Sort heap threshold (4 KB)	Double the value allocated for SORTHEAP. See Table 11 on page 108.
ASLHEAPSZ	Application support layer heap size (4 KB)	15
MON_HEAP_SZ	Database monitor heap size (4 KB)	128 (minimum)
UDF_MEM_SZ	UDF shared memory set size (4 KB)	256
RQRIOBLK	Max. requester I/O block size (bytes)	65535
QUERY_HEAP_SZ	Query heap size (4 KB)	16384
KEEPDARI	Keep DARI process	YES
QUERY_HEAP_SZ	Query heap size (4 KB)	16384
MAX_COORDAGENTS	Maximum number of coordinating agents	MAXAGENTS

db2set Parameters

Use the db2set command to set the parameters (for example, db2set DB2_RR_TO_RS = YES) referenced in Table 10. (Under Windows, you would access this through the DB2 Command Window, accessible from the DB2 for Windows 2000 client.)

Table 10. db2set Parameters

Parameter	Explanation	Setting
DB2_RR_TO_RS	Improves DB2 performance with Siebel eBusiness Applications. *Set to YES only in production environment servers.*	YES
DB2_MMAP_WRITE	Recommended setting only; you should evaluate this setting for your particular configuration and environment.	OFF

Table 10. db2set Parameters

Parameter	Explanation	Setting
`DB2_MMAP_READ`	Recommended setting only; you should evaluate this setting for your particular configuration and environment.	`OFF`
`DB2_CORRELATED_PREDICATES`	When set to ON, the optimizer is able to determine whether predicates in a query are related. This permits DB2 to calculate the filter factor more accurately.	`ON`
`DB2_INDEX_2BYTEVARLEN`	This parameter must always be set to ON. Otherwise, you will not be able to create indexes with columns greater than 255 bytes.	`ON`
`DB2_PIPELINED_PLANS`	Tells the DB2 optimizer to favor pipeline execution plans; in other words, plans which are left deep and have no temporary result sets.	ON
`DB2_INTERESTING_KEYS`	Limits the number of execution plans generated by the DB2 optimizer.	ON
`DB2_PARALLEL_IO`	Useful when using RAID devices. For more information, refer to relevant IBM documentation.	ON
`DB2_STRIPED_CONTAINERS`	Useful when using RAID devices. For more information, refer to relevant IBM documentation.	ON

NOTE: After changing any of these settings, perform a `db2stop/db2start` to implement the changes in your DB2 database.

DB2 Database Configuration Parameters

The database configuration parameters can be set using the `update database configuration` command of the DB2 Command Line Processor or using the DB2 Control Center. See the IBM DB2 technical documentation for more information on modifying the database configuration parameters.

Set the parameters in Table 11 for *each* database within an instance on which you run your Siebel eBusiness Applications. For other parameters of the same type, accept the default settings.

Table 11. DB2 Database Configuration Parameters

Parameter	Explanation	Setting
`DFT_DEGREE`	Degree of parallelism (1=turn query parallelism off)	`1`
`DFT_QUERYOPT`	Default query optimization class	`3`
`DBHEAP`	Database heap (4 KB)	`7429`
`CATALOGCACHE_SZ`	Catalog cache size (4 KB)	`5558`
`LOGBUFSZ`	Log buffer size (4 KB)	`512` (For Windows, set this to `256`.)
`UTIL_HEAP_SZ`	Utilities heap size (4 KB)	`5000`
`LOCKLIST`	Maximum storage for lock list (4 KB)	`5000` (This is the minimum setting. It may be increased.)
`APP_CTL_HEAP_SZ`	Maximum applications control heap size (4 KB)	`900` (Recommended size may increase or decrease with the number of users.)
`STMTHEAP`	SQL statement heap (4 KB)	`8192`
`SORTHEAP`	Sort list heap (4 KB)	`20,000—40,000` Recommended size; this may increase or decrease depending on the amount of memory in the database server machine and the size of the data. A 20,000 setting allows `SORTHEAP` to increase up to 80 MB.
`APPLHEAPSZ`	Default application heap (4 KB)	`2500` (Recommended size may increase or decrease with the number of users.)
`STAT_HEAP_SZ`	Statistics heap size (4 KB)	`8000`
`MAXLOCKS`	Percentage of lock lists per application	`5`
`LOCKTIMEOUT`	Lock timeout (sec.)	`90` to `150`
`CHNGPGS_THRESH`	Changed pages threshold	`5`
`NUM_IOCLEANERS`	Number of asynchronous page cleaners	Number of CPUs

Table 11. DB2 Database Configuration Parameters

Parameter	Explanation	Setting
INDEXSORT	Index sort flag	YES
SEQDETECT	Sequential detect flag	YES
LOGRETAIN	Sequential or circular log files	NO Setting this parameter to YES means that log files will be archived and the potential exists for the file system contained the log files to fill up if you do not move or archive the logs.
AVG_APPLS	Average number of active applications	1
MAXFILOP	Maximum DB files open per application	500
LOGFILSIZ	Log file size (4 KB)	Development environments: 8000-16000 Production environments: 64000
LOGPRIMARY	Number of primary log files	25-50 The value of LOGPRIMARY and LOGSECOND together may not exceed 128.
LOGSECOND	Number of secondary log files	Accept the DB2 UDB default value; increase this value if secondary log files are required for your deployment.
SOFTMAX	Triggers bufferpool flushing	50
NUM_IOSERVERS	Number of disks on which the database resides	Number of disks

Installing the Stored Procedures and User-Defined Functions

NOTE: If you are performing an upgrade from Release 7.0.x to Release 7.5, skip this procedure. You already completed this step during your upgrade to Release 7.0.x.

If you are upgrading from Release 5.x or 6.x and your RDBMS is DB2 UDB, you must install the stored procedures and user-defined functions (UDFs) on the database server. To do this, you must first transfer them to the database server, and have installed the database server components. (For information on installing database server components, refer to the chapter on installing the Siebel Database Server for DB2 Universal Database in *Siebel Server Installation Guide* for the operating system you are using.)

If you use a different RDBMS platform, skip to "Preparing Application Data for Upgrade" on page 127.

The user-defined functions (UDFs) and stored procedures must be transferred to and installed on the database server to support the Siebel product. Any method that transfers the necessary files to the correct location on the database server is acceptable.

NOTE: Compiling stored procedures in DB2 creates binary files. You must copy these files to the database server machine's `sqllib\function` directory.

To copy and install the stored procedure code, follow the procedure appropriate to your operating system. You first must delete the old store procedures. Then you need to install the Siebel stored procedure library on the DB2 database server host.

Before you perform this procedure, you might need to upgrade your DB2 database software. You might also be required to apply a DB2 fix pack. See *Siebel System Requirements and Supported Platforms* for more information about required versions.

To delete the old stored procedures

- Delete the old stored procedures from the `sqllib\function` directory:

  - On the Siebel Database Server on the Windows platform:

    `nextseq.dll`

    `nextseqd.exe`

    `siebstat.dll`

    `siebsdmy.dll`

    `siebtrun.dll`

    `siebfcst.dll`

    `mapping.dll`

    `updt2col.dll`

    `updt_addr_org_sp.dll`

    `updt_addrorg_intsp.dll`

    `updt_addr_per_sp.dll`

- On the Siebel Database Server on UNIX:

  nextseq

  nextseqd

  siebstat

  siebupg1

  siebsdmy

  mapping

  updt2col

  siebtrun

  siebfcst

  updt_addr_org_sp

  updt_addr_per_sp

  updt_addrorg_intsp

To copy and install the stored procedure code

1. Install the Siebel stored procedure library on the DB2 database server host.
2. Log onto the Siebel source installation machine, and navigate to the source installation subdirectory that contains the Siebel Database installation objects.

   The directory that contains the file to install (siebproc) is

   $*SIEBEL_ROOT*/db2udb/siebproc/DBSRVR_OS

   where:

   DBSRVR_OS = the operating system your database server runs on; for example, aix.
3. Put the siebproc file (on UNIX this is called siebproc) into the FUNCTION subdirectory within the DB2 UDB instance directory (where DB2 UDB is installed) on the Siebel Database Server.

   For example, on AIX, this location might be $*INST_HOME*/sqllib/function.

Preparing the Database for a DB2 Upgrade

You need to perform the following procedures to prepare your database for a DB2 UDB upgrade:

If you are upgrading from Release...	Perform the following procedure...
5.x, 6.x, or 7.0.x	Create DB2 temporary tablespaces and bufferpools.
5.x, 6.x, or 7.0.x	Increase 4-KB tablespaces.
5.x	Create 16-KB tablespaces and bufferpools.
6.x or 7.0.x	Increase 16-KB tablespaces.
5.x or 6.x	Create 32-KB tablespaces and bufferpools.
7.0.x	Increase 32-KB tablespaces and bufferpools.
5.x, 6.x, or 7.0.x	Determine tablespace page size requirements.
5.x, 6.x, or 7.0.x	Verify that the Application Development Client/Tool is installed on your database server.

Creating DB2 Temporary Tablespaces and Bufferpools

If your RDBMS is DB2 UDB, verify that you have 16-KB and 32-KB temporary tablespaces to use for sorting and other SQL processing. Both the 16-KB and 32-KB temporary tablespaces require dedicated bufferpools.

To create a 16-KB temporary tablespace

1. Create a 16-KB bufferpool with at least 5,000 16-KB pages.
2. Create a 16-KB temporary tablespace as system managed space (SMS) that can be expanded to 2 GB of storage.

To create a 32-KB temporary tablespace

1. Create a 32-KB bufferpool with at least 1,000 32-KB pages.
2. Create a 32-KB temporary tablespace as SMS that can be expanded to 2 GB of storage.

Increasing DB2 4-KB Tablespace

If you are upgrading from Release 5.x, 6.x, or 7.0.x and your RDBMS is DB2 UDB, you must increase the size of your 4-KB tablespace.

To increase the size of your 4-KB tablespace

- Increase the 4-KB tablespace using the DB2 tools of your choice. For sizing requirements, see Table 11 on page 108.

This completes the tablespace allocation for your 4-KB tablespace.

Creating DB2 16-KB Tablespace and Bufferpool

If you are upgrading from Release 5.x and your RDBMS is DB2 UDB, create a new 16-KB tablespace and bufferpool, to which the data in your previous installation's 8-KB tablespace will be migrated.

NOTE: Skip this step if you are upgrading from Release 6.x or 7.0.x, because you already created your 16-KB tablespace and bufferpool. Instead, perform the task described in "Increasing DB2 16-KB TableSpace."

To create 16-KB tablespace and bufferpool

1 Create a 16-KB bufferpool, with a number of pages equal to at least twice the number of pages in the 8-KB bufferpool in your previous installation.

> NOTE: If your system has limited RAM, you may want to reduce the size of your current 8-KB bufferpool or increase the size of your swap file before upgrading; this will prevent an upgrade failure due to lack of memory.

2 Create a 16-KB tablespace using the DB2 tools of your choice.

This completes creation of your new 16-KB tablespace and bufferpool.

Increasing DB2 16-KB TableSpace

If you are upgrading from Release 6.x or 7.0.x and your RDBMS is DB2 UDB, you must increase the size of your 16-KB tablespace.

NOTE: Skip this step if you are upgrading from Release 5.x.

To increase the size of your 16-KB tablespace

- Increase the 16-KB tablespace using the DB2 tools of your choice. For sizing requirements, see Table 11 on page 108.

This completes the tablespace allocation for your 16-KB tablespace.

Creating DB2 32-KB Tablespace and Bufferpool

If you are upgrading from Release 5.x or 6.x and your RDBMS is DB2 UDB, you must create a new 32–KB tablespace and bufferpool in order for the upgrade to complete successfully.

NOTE: Skip this step if you are upgrading from Release 7.0.x (for example, Release 7.0.3 or 7.0.4), because you already created your 32-KB tablespace and bufferpool. Instead, perform the task described in "Increasing DB2 32-KB Tablespace" on page 115.

To create 32-KB tablespace and bufferpool

1. Create a 32-KB bufferpool, with the number of pages equal to at least 1,000 32-KB pages.
2. Create a 32-KB tablespace using the DB2 tools of your choice.

This completes creation of your new 32-KB tablespace and bufferpool.

Increasing DB2 32-KB Tablespace

If you are upgrading from Release 7.0.x and your RDBMS is DB2 UDB, you must increase the size of your 32-KB tablespace.

NOTE: Skip this step if you are upgrading from Release 5.x or 6.x.

To increase the size of your 32-KB tablespace

- Increase the 32-KB tablespace using the DB2 tools of your choice. For sizing requirements, see Table 11 on page 108.

This completes the tablespace allocation for your 32-KB tablespace.

Determining Tablespace Page Size Requirements for DB2 UDB

There are four standard database managed tablespaces (DMS) that hold Siebel tables and indexes—a 4-KB, 16-KB, 32-KB tablespace, for various sized tables, and a tablespace to hold indexes. However, in some cases, you may have Siebel tables in custom tablespaces.

If a custom table has an estimated page size greater than its current tablespace page size, it will not fit in its tablespace after the upgrade, and the upgrade will fail. The following utility will determine whether each of your current custom tables will increase in size to the point that it will require a move to a larger tablespace.

This utility must be run prior to the upgrade, before any steps of the upgrade are executed.

To determine tablespace page size requirements for DB2 UDB

1. From the `siebsrvr/bin` directory, type the following command line:

   ```
   tblsize /U $Tableowner /P $Password /C $ODBCDataSource /F
   $DDLFilename /B $DefaultTablespace /X $DefaultIndexspace /K
   $16Ktablespace /V $32Ktablespace /Q $ReportFilename /L
   $LogFilename
   ```

   where:

   - `Tableowner` = Tableowner
   - `Password` = Tableowner password

- `ODBCDataSource` = Data source of the database
- `DDLFilename` = Name of the DDL file (This file is called `ddl.ctl`, and it is located in the `dbsrvr/db2udb` directory.)
- `DefaultTablespace` = Name of the 4-KB page standard Siebel tablespace
- `DefaultIndexspace` = Name of the standard Siebel index space
- `16Ktablespace` =Name of the 16-KB page standard Siebel tablespace
- `32Ktablespace` = Name of the 32-KB page standard Siebel tablespace
- `ReportFilename` =Name of the report generated by the utility
- `LogFilename` = Name of the log file (The default name is `custtbl.log`.)

2 Review the report generated by the utility to determine if the estimated table pagesize postupgrade is larger than the size of the actual custom table pagesize.

An example of the report generated by this utility is provided below:

```
Table Name = S_EVT_ACT

Custom Tablespace Id = 5

Custom Tablespace Name = CUST_TBS_EVT_ACT

Custom Tablespace Pagesize = 4096

Estimated Table Pagesize (postupgrade) = 5067

Status = Does not fit in its custom tablespace
```

3 For each table that has `Status: Does not fit in its custom tablespace`, you must create a larger custom tablespace which is larger than the estimated table pagesize postupgrade.

4 Move the tables from their old tablespaces to the new ones by running `ddlmove`.

`ddlmove` is a utility for moving tables from one tablespace to another tablespace. This utility is located under the `siebsrvr/bin` directory.

To run `ddlmove`, submit the following arguments:

```
ddlmove /U $Tableowner /P $TablePassword /C $ODBCDataSource /E
$Stop_on_DDL_Error /G $Grantee /B $Tablespace /X $IndexTablespace
/M $TableName /L $LogFilename /Z $UCS2Database
```

where:

- `Tableowner` = Tableowner of the database (Required)
- `TablePassword` = Password of the tableowner of the database (Required)
- `ODBCDataSource` = Data source of the database (Default environment variable: SIEBEL_DATA_SOURCE)
- `Stop_on_DDL_Error` = Stop on DDL Error (Default: Y)
- `Grantee` = Grantee for tables
- `Tablespace` = Name of the tablespace that you are moving the table to
- `IndexTablespace` = Name of the index space that you are moving the table to
- `TableName` = Table Name Like Support (Default: N)
- `LogFilename` = Name of the log file (The default name is ddlmove.log.)
- `UCS2Database` = (Default: N)

NOTE: If there are problems reported by the sizing utility, you must resolve the tablespace page sizes before you proceed with the upgrade.

Verifying Installation of the DB2 UDB Application Development Client

If you are upgrading to Release 7.5 from Release 5.x, 6.x, or 7.0.x and your RDBMS is DB2 UDB, you must verify that the DB2 UDB Application Development Client is installed on your Siebel Database Server before proceeding with the upgrade. To do this, navigate to the appropriate directory on the Siebel Database Server and verify that the DB2 UDB Application Development Client is installed.

Table 12 lists the DB2 UDB Application Development Client components that must be installed on your Siebel Database Server depending on your platform. Use this table to verify that you have the correct components installed.

Table 12. DB2 UDB Application Development Client Components

Platform	DB2 UDB Application Development Client Components
Windows	DB2 Application Development Client
AIX	db2_07_01.adt.rte 7.1.0.xx COMMITTED Application Development Tools db2_07_01.adt.samples 7.1.0.xx COMMITTED ADT Sample Programs
HP	DB2V7SKL 7.1.0.40 Application Development Tools for HP-UX
Solaris	application db2adt71 Application Development Tools (ADT) (PTF 1720500-00) application db2adts71 ADT Sample Programs (PTF 1720500-001)

If the DB2 Application Development Client is not installed, you must install it. For more information, refer to the relevant IBM documentation.

Preparing Tables for Upgrade

Take the following measures to prepare your tables for the upgrade.

Preserving Custom Indexes on Tables

Consider the following implications and carefully plan your upgrade to preserve custom indexes.

- **Custom indexes against extension columns on obsolete tables.** Careful planning is necessary if you have custom indexes against extension columns on obsolete tables. You must migrate data to new extension columns before you upgrade the Siebel database schema. For assistance, contact Siebel Technical Support.
- **Custom indexes that were not defined through Siebel Tools in the Siebel repository.** If your DBA created custom indexes in your current schema that were not defined through Siebel Tools in the Siebel repository, these indexes will be dropped during the upgrade process.
- **Custom indexes on interface tables.** Custom indexes on interface tables are not recreated during the upgrade, but they can be restored when the upgrade is complete.
- **Custom indexes on base tables.** Release 7.x upgrade automatically drops and recreates custom indexes on base tables.

NOTE: Custom indexes may need to be changed to reflect schema changes. You should reevaluate custom indexes for applicability in the new release.

For more information about applying custom indexes, see *Siebel Tools Reference*.

Considerations for Clustered Indexes

If you created clustered indexes on base tables and Release 7.x introduces a different clustered index on the same table, the upgrade process will recreate custom indexes as nonclustered and create the Siebel index as clustered.

For IBM DB2 UDB, indexes that reside on tables that have been defined with append mode enabled will be recreated as nonclustered indexes during the upgrade. Tables that have been created with append mode enabled cannot contain clustered indexes.

Disabling Customized Triggers

Release 7.x does not support customized triggers. If you have created customized triggers on your Siebel base tables, disable them before you perform the upgrade. You will then need to recreate them after the upgrade is finished.

Dropping Customized Views

If you have created customized views on your Siebel base tables, you must drop them before you perform the upgrade. If they are still applicable after the upgrade, you will need to recreate them after the upgrade is finished.

Identifying DB2 UDB Long Columns for Truncation

In Release 7.5, the maximum length for DB2 UDB long columns with a type of varchar has reduced to 16350 from 16383. Long columns of type varchar that exceed 16,350 will be truncated.

To prevent a data truncation error that may cause transaction processing (`txnproc`) or transaction routing (`txnroute`) to fail, perform the following steps to identify these columns and reduce the data in these columns.

To identify and reduce the length of long varchar columns

1 From any shell, open the script `chk16350.ksh`, and edit the following parameters as appropriate for your deployment:

   `SRC_USR` = username of the source database

   `SRC_PSWD` = password for the source database

   `SRC_TBLO` = tableowner of the source database

   `SRC_TBLO_PSWD` = tableowner password for the source database

   `SRC_ODBC` = ODBC data source name of the source database (edit the value "`CHANGE_ME`")

   `SRC_REPOSITORY_NAME` = repository name of the source database

   `DBSRVR_ROOT` = directory where you installed the dbsrvr component of the Siebel Server (edit the value "`CHANGE_ME`")

   `SIEBEL_ROOT` = directory where you installed the siebsrvr component of the Siebel Server (edit the value "`CHANGE_ME`")

   `VALID_RESULTS_DIR` = directory where you want the output files to be generated (edit the value "`CHANGE_ME`"); this must be an existing directory

   This script produces two files:

   - **long_trunc_cols.rpt.** This report identifies all long varchar columns that are longer than 16350 characters.
   - **update_trunc.sql.** This SQL file will generate update statements that truncate identified columns to 16350 characters.

2 Reduce the data in these columns using either of the following methods:

   - Manually review the columns in the `long_trunc_cols.rpt` report and manually reduce the size of each column identified.

- Run `update_trunc.sql` using the DB2 command line processor.

CAUTION: If you do not truncate or otherwise reduce the data in these columns, you will receive a “data truncated” error, and transaction processing and transaction routing may fail.

Updating Statistics on DB2 UDB

DB2 UDB databases use statistics about tables and indexes to compute the most efficient access plans. When statistics become inaccurate, which can happen for tables with high insertion rates, high deletion rates, or both, and for associated indexes, the performance of database operations can degrade dramatically.

To update statistics, run `updatestats.sql` or use DB2 UDB specific statements to refresh statistical information in the Siebel database.

Preparing the Siebel Repositories for Upgrade

Take the following measures to prepare your development or production environment repositories for upgrade.

Preparing Development Environment Repositories

Four separate repositories are used during the development upgrade process:

- **Your existing development repository.** To prevent a naming conflict, before you run the upgrade, rename your existing development repository (*Siebel Repository*) to *Prior Customer Repository.* After the upgrade, your new Release 7.5 development repository will be given the name *Siebel Repository.*
- **Three new repositories.** These are automatically loaded when you run the upgrade wizard:
  - Prior standard repository (Release 5.x or 6.x)
  - New standard repository (Release 7.5)
  - New customer repository (Release 7.5), which after the merge, will become the customized 7.5 repository.

Verify Repository Names

Before these new repositories are loaded, verify that your existing repositories do not use the names reserved for the upgrade process:

- **New Customer Repository.** Your new custom repository, that results from the merge and includes your custom configurations.
- **New Siebel Repository.** The new Siebel standard repository.
- **Prior v5.x Siebel Repository** (for upgrades from Release 5.x).
- **Prior v6.x Siebel Repository** (for upgrades from Release 6.x).
- **Prior v7.x Siebel Repository** (for upgrades from Release 7.x).

Checking In Archived Release 5.x and Release 6.x Repository Objects

If you archived Release 5.x and Release 6.x repository objects as .sif files that you want to have available in your Release 7.5 application, you need to check these objects back into the repository before you begin your upgrade. If you do not check these objects back into the repository, they will not be upgraded.

NOTE: You only need to check in those archived objects that you will need in the future and want to have available in your upgraded application.

Renaming Your Existing Development Repository

Use the version of Siebel Tools that matches your old version of Siebel eBusiness Applications (for example, Siebel Tools Release 6.x for upgrades from Release 6.x). Connect directly to the development database server, and then perform the following steps to make sure that your existing repositories follow the correct naming conventions.

To rename the repository

1. Using the appropriate prior version of Siebel Tools, connect to the Siebel Database Server.
2. Change the name of the existing repository, as described below:
    a. In the Object Explorer, click the Types tab.
    b. Click Repository.
    c. In the Repository view, click Name.
    d. Locate the appropriate repository in the list applet and rename it `Prior Customer Repository`.
3. Step off the list to commit the record to the database.

The upgrade process will verify the repository names. If no repository is named Prior Customer Repository, the Siebel Repository will be renamed to Prior Customer Repository in the target database, so that the upgrade will execute properly.

NOTE: Do not rename repositories with a reserved name appended by additional characters. This will result in an error. Instead, prefix the name with additional characters.

For more information about renaming repositories, refer to *Siebel Tools Reference.*

Preparing Production Environment Repositories

Two separate repositories are used during the production upgrade process:

- **Your existing production repository.** To prevent a naming conflict, before you run the upgrade, rename your existing production repository (*Siebel Repository*) to *Prior Customer Repository.* After the upgrade, your new Release 7.5 production repository will be given the name *Siebel Repository.*
- **New Customer Repository.** This repository will be loaded when you run the upgrade wizard.

Renaming Your Existing Production Repository

Use the version of Siebel Tools that matches your old version of Siebel eBusiness Applications (for example, Siebel Tools Release 6.x for upgrades from Release 6.x). Connect directly to the production database server, and then perform the following steps to make sure that your existing repositories follow the correct naming conventions:

To rename the repository

1 Using the appropriate prior version of Siebel Tools, connect to the Siebel Database Server.

2 Change the name of the existing repository, as described below:

  a In the Object Explorer, click the Types tab.

  b Click Repository.

  c In the Repository view, click Name.

d. Locate the appropriate repository in the list applet and rename it `Prior Customer Repository`.

3. Step off the list to commit the record to the database

NOTE: The upgrade process will verify the repository names. If no repository is named Prior Customer Repository, the Siebel Repository will be renamed to Prior Customer Repository in the target database so that the upgrade will execute properly.

If you need more information about renaming repositories, refer to *Siebel Tools Reference.*

Recording Dock Objects and Visibility Rules

NOTE: If you are performing a production environment upgrade and you already performed this procedure for a development environment upgrade, you do not need to repeat it.

Changes to visibility rules and dock objects require the assistance of Expert Services.

Modified visibility rules will be dropped during the upgrade. Manually record your changes to dock object visibility rules so you can evaluate the need to reapply the changes after the upgrade is complete.

Dock objects and visibility rules created by using Docking Wizard will be preserved unless they become invalid after the upgrade. Manually record any changes that you made through the Docking Wizard so that you can evaluate the need to reapply the changes after the upgrade is complete.

NOTE: Changes to visibility rules require the assistance of Expert Services.

Preparing Application Data for Upgrade

If you use one of the following Siebel eBusiness Applications, you need to perform preupgrade procedures to prepare your data for upgrade to Release 7.x:

- Siebel eChannel
- Siebel eConfigurator
- Forecasting
- Quotes
- Timesheet
- Calendar
- Siebel Financial Services

Siebel eChannel

NOTE: If you are upgrading from Release 7.0.x to Release 7.5, you do not need to perform this procedure.

Before upgrading to Release 7.x, if you have modeled your partners as both Accounts and Divisions (or Organizations) you should merge these two records to make sure that only one record will exist for each partner company in the new single party model. Similarly, if you have modeled partner employees as Contacts and Employees, you should merge these records to result in a one-person record in the upgraded database. For assistance with the procedure to identify and merge matching records, contact Siebel Professional Services.

Siebel eConfigurator

NOTE: If you are upgrading from Release 7.0.x to Release 7.5, you do not need to perform this procedure.

The Siebel Configurator's data model migration is embedded in the Siebel standard upgrade process. However, in order to make sure that the model is upgraded properly, customers will need to have appropriate model product associations. For more information on this and other recommended preupgrade procedures, refer to *Siebel Interactive Selling Applications Upgrade Guide*.

Forecasting

NOTE: If you are upgrading from Release 7.0.x to Release 7.5, you do not need to perform this procedure.

Before upgrading from a previous version of Siebel eBusiness Applications, system administrators should decide whether or not the revenues for non-primary sales team members should be upgraded. They may want to consider upgrading non-primary sales team members' revenues if their organization:

- Uses non-primary forecasting and wants to continue using a similar non-primary forecasting approach. (Running queries on the Revenue table may not be adequate to meet this need.)
- Wants each sales team member to have a different opinion on the close date and revenue amount for each opportunity.
- Wants to assign, to each sales team member, some portion of each opportunity that is not related to a specific product or some percentage of the overall opportunity value.

For each opportunity in previous versions of Siebel eBusiness Applications, the automatic upgrade evaluates the opportunity sales team records for non-primary sales team members. Each sales team member record is used to create a revenue record in Release 7.x, if the following is true.

- The primary flag is not checked.
- The revenue amount is nonzero.
- The commit flag is checked.

Before upgrading, make sure that your opportunity sales team records are appropriately marked as committed or not committed. Records marked as committed will be automatically upgraded; records marked as not committed will not be upgraded. Most system administrators will not need to upgrade uncommitted sales team records.

Quotes

NOTE: If you are upgrading from Release 7.0.x to Release 7.5, you do not need to perform this procedure.

Prior to upgrading to Release 7.x, you need to determine which of your Release 6.x configuration solutions are to be upgraded to Customizable Product quote items in Release 7.x. For products that are being upgraded as Customizable Products, you should make sure that the Model Product field in Product Administration is checked. This will allow these products to be upgraded as Customizable Products in Release 7.x. For more information, see *Siebel Interactive Selling Applications Upgrade Guide*.

For products that do not need to be upgraded as Customizable Products, the Model Product field should be unchecked. This will also make sure quotes using these products as solutions in 6.x will be upgraded as Packaged products in Release 7.x. All configuration model associations for these products will be removed and these products will behave like hierarchical product bundles. For more information on Packaged products and eConfigurator, see *Product Administration Guide* and *Siebel Interactive Selling Applications Upgrade Guide*.

In Siebel Quotes 6.x, the Discount Amount field on a line item was automatically populated to `$0.00`. This meant that, for any line item, users had to clear the discount amount field if they wanted to apply a header level discount after adding a line item.

If you implemented Siebel Quotes 6.x so that the Discount Amount field was read the same whether it was `NULL` or `$0.00`, thereby allowing users to enter a header discount after adding a line item, the situation described above did not occur.

In Release 7.x, this issue is no longer applicable. The Discount Amount field is left as `NULL` unless a user explicitly enters a value, including `$0.00`. Therefore, when a you upgrade from Release 6.x to Release 7.x, you need to determine whether no change had been made in Siebel Quotes 6.x to Discount Amount (thus letting it be automatically populated to `$0.00`) or whether Siebel Quotes 6.x had been modified to read `NULL` or `$0.00` in Discount Amount as the same. In the event the latter is true, you need to run a Quote script with the following SQL statement applied before beginning the upgrade, that is, before running the configuration utility and applying it to development and production upgrades.

```
update S_QUOTE_ITEM

set   DISCNT_AMT = NULL

where  DISCNT_AMT = 0
```

This script will account for the fact that the Discount Amount field is read the same whether it is `NULL` or `$0.00.` Administrators running Siebel Quotes 6.x without modifications to Discount Amount can ignore this script.

Timesheet

If you are upgrading from a pre-7.5 version of Siebel eBusiness Applications, please refer to Siebel SupportWeb for a technical note about Time Sheet Conversion.

Siebel Financial Services

NOTE: If you are upgrading from Release 7.0.x to Release 7.5, you do not need to perform this procedure.

This section provides additional preupgrade tasks for Siebel Financial Services customers upgrading from Releases 5.x and 6.x to Release 7.x of Siebel Financial Services. The tasks described in this section are unique to Siebel Financial Services and should be performed in addition to the preupgrade tasks for the development environment provided in the previous sections of this chapter.

Verifying Household Data Integrity

The household verification utility is unique to Siebel Financial Services. If you have chosen to implement the new household design, you must run this utility prior to the upgrade.

CAUTION: If you have elected not to upgrade to the new household design, do not run `HH_Mig_populate.sql`, and make sure that the temporary table `TEMP_HH_OBJ_MIG` does not exist in the database.

When you migrate to the new household design, direct relationships between a household and its entities (Policy/Financial Accounts, Activities, Claims, Opportunities, Service Requests, and Companies) become obsolete. Therefore, you must run the household verification utility before the upgrade to verify data integrity. After the upgrade, you must be able to see greater than or equal to the amount of data that you had in Release 6.x.

The household verification script, `HH_Mig_populate.sql`, verifies that at least the same number of entities will belong to a household after the upgrade as belonged to it in your previous release. This script is located in *`$SIEBEL_ROOT/dbsrvr/database_platform`* directory.

The household verification utility operates with the following assumptions:

- A household has at least one contact.
- The primary contact of a Policy/Financial Account is one of the contacts associated with this Policy/Financial Account.
- The primary contact of a Claim is one of the contacts associated with this Claim.
- The primary contact of an Opportunity is one of the contacts associated with this Opportunity.
- The primary contact of a Company is one of the contacts associated with this Company.

To run the household verification utility

- Type the following command :

  ```
  odbcsql /U $Tableowner /P $Password /S $ODBCDataSource /a /c
  REM /separator / /O $OutputFileLocation/HH_Mig_populate.txt
  /L $LogFileLocation/HH_Mig_populate.log $ScriptLocation/
  HH_Mig_populate.sql  /v
  ```

  where:

  `Tableowner` = Tableowner

  `Password` = Tableowner password

  `ODBCDataSource` = Data source of the database

  `OutputFileLocation` = Location and name of the output file (`$`*`SIEBEL_ROOT`*`/siebsrvr/Log/HH_Mig_populate.txt`)

  `LogFileLocation` = Location and name of the log file (`$`*`SIEBEL_ROOT`*`/siebsrvr/Log/HH_Mig_populate.log`)

  `ScriptLocation` = Location and name of the script (`$SIEBEL_ROOT/dbsrvr/database_platform/HH_Mig_populate.sql`)

  Example:

  ```
  odbcsql /U $Tableowner /P $Password /S $ODBCDataSource /a /c
  REM /separator / /O /$SIEBEL_ROOT/siebsrvr/Log/
  HH_Mig_populate.txt /L /$SIEBEL_ROOT/siebsrvr/Log/
  HH_Mig_populate.log /$SIEBEL_ROOT/dbsrvr/database_platform/
  HH_Mig_populate.sql /v
  ```

  The program populates a temporary table with data, `TEMP_HH_OBJ_MIG`. When the program has finished running, it generates a report based on your designated output file; for example, `HH_Mig_populate.txt`.

  Output is in the form of row IDs.

  - If there is no output, you do not need to take any action. Data integrity is verified between Release 6.x and 7.x.

- If you do receive an output, this means that no contact is associated with the household for a particular entity. This can occur in the following scenarios:
  - Contact is correct and household is incorrect.
  - Contact is incorrect and household is correct.
  - Contact is incorrect and household is incorrect.

  At least one contact associated with an entity must also be associated with the household for that entity.

Preparing Application Tables for Upgrade

DESC_TEXT is a column that resides on two tables, S_CONDITION and S_PROD_BNFT. To broaden platform support, this column will be reduced in length from 250 to 200 during the upgrade from Release 6.x to Release 7.x. Therefore, before you run the upgrade, examine these two tables for records that contain more than 200 characters. If you find oversized records, manually reduce the size of these records to 200 characters or less.

Review the following tables:

- S_CONDITION
- S_PROD_BNFT

To prevent loss of data, perform the following procedure before you upgrade to Release 7.x.

To prepare S_CONDITION and S_PROD_BNFT for upgrade

1. Review records in S_CONDITION for records that contain more than 200 characters.
2. For records which contain more than 200 characters, edit the records so that their row length is less than 200.
3. Repeat this procedure for S_PROD_BNFT.

Continuing with the Upgrade

Follow the instructions for the environment you are upgrading:

- **Development Environments.** If you completed preupgrade tasks for the development environment, go to Chapter 4, “Upgrading the Siebel eBusiness Application,” and follow the instructions for a development environment upgrade. Skip procedures that only apply to production environments.
- **Production Environments.** If you completed preupgrade tasks for the production environment, go to Chapter 4, “Upgrading the Siebel eBusiness Application,” and follow the instructions for a production environment upgrade. Skip procedures that only apply to development environments.

Upgrading the Siebel eBusiness Application 4

This chapter describes the tasks involved in upgrading your Siebel eBusiness Application. To better understand which phase of the upgrade process you are in and what tasks still remain, refer to Figure 6. Figure 6 illustrates where you are in the upgrade process depending on whether you are performing a development or production environment upgrade.

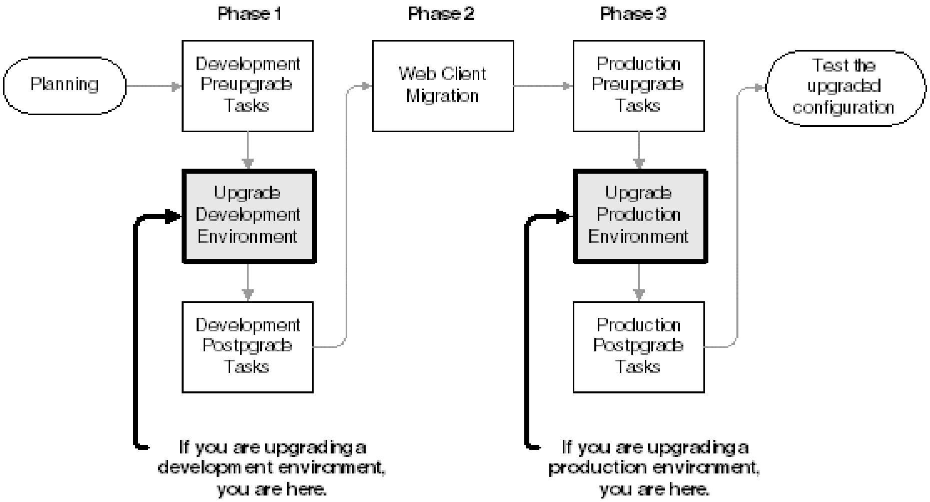

Phase 1
Phase 2
Phase 3
Planning
Development Preupgrade Tasks
Web Client Migration
Production Preupgrade Tasks
Test the upgraded configuration
Upgrade Development Environment
Upgrade Production Environment
Development Postpgrade Tasks
Production Postpgrade Tasks
If you are upgrading a development environment, you are here.
If you are upgrading a production environment, you are here.

Figure 6. You Are Here

Do not proceed unless you have completed the following procedures, as appropriate for your upgrade.

If you are upgrading a development environment

- Complete the tasks described in "Preupgrade Tasks" to prepare your development environment data and repositories for upgrade.

If you are upgrading a production environment

- Complete preupgrade, upgrade, and postupgrade tasks for your development environment.

  NOTE: If you do not have a development environment because you did not customize the Siebel eBusiness Applications, go to Appendix B to upgrade a test environment before you upgrade your production environment.

- Thoroughly test the application on your development or test environment.
- Complete migration procedures described in Chapter 6, “Migrating the Client Configuration to the Web Client” if you have customized applets or views, scripts on user-interface objects, or client-side interfaces to migrate from the Windows Client to the Web Client.

A checklist of upgrade tasks is provided in Table 13. Perform the procedures for the environment that you are upgrading.

NOTE: If a procedure is specific to one environment, this will be indicated by a caution at the beginning of the procedure; for example, CAUTION: This procedure only applies to development environment upgrades.

Table 13. Upgrade Tasks

Upgrade Task		Environment	
		Dev.	Prod.
1	Install Siebel Tools on development workstations. See "Installing Siebel Tools on the Development Workstations" on page 138.	✔	
2	Upgrade the Siebel Gateway and Siebel Servers. See "Upgrading Siebel Gateway and Siebel Servers" on page 140.	✔	✔
3	Install the Siebel Database Server Software. See "Installing the Siebel Database Server Software" on page 143.	✔	✔
4	Upgrade Siebel database schema. See "Upgrading the Siebel Database Schema" on page 147.	✔	✔
5	Add new license keys. See "Adding New License Keys" on page 159.	✔	
6	Prepare the prior customer repository for merge. See "Preparing the Prior Customer Repository for the Merge" on page 160.	✔	
7	Migrate strings, merge labels and fields, and merge templates. See "Migrate Strings, Merge Labels and Fields, and Merge Templates" on page 161.	✔	
8	Upgrade copied objects. See "Automatic Upgrade of Copied Objects" on page 165.	✔	
9	Perform configuration steps for upgrade inheritance. See "Configuration Steps for Upgrade Inheritance" on page 169.	✔	
10	Merge the repository. See "Performing a Repository Merge" on page 175.	✔	
11	Generate EIM temporary columns. See "Generating EIM Temporary Columns" on page 187.	✔	
12	Upgrade your custom database schema. See "Upgrading the Custom Database Schema" on page 189.	✔	✔
13	Add new license keys. See "Adding New License Keys" on page 159.		✔

Installing Siebel Tools on the Development Workstations

CAUTION: This procedure only applies to development environment upgrades. If you are upgrading a production environment, skip to "Upgrading Siebel Gateway and Siebel Servers."

The upgrade process requires the Siebel Tools application. If you do not have Siebel Tools, you need to contact Siebel technical support for assistance. Siebel Tools is required even if you have no customizations.

To install Siebel Tools

1. Remove all but one installation of the previous version of Siebel Tools from all development workstations.

   - For upgrades from version 5.x, double-click the UnInstall icon in the Siebel Tools program folder.

   - For upgrades from 6.x, navigate to the Control Panel, and double–click Add/ Remove Programs. Select all but one installation of the previous version of Siebel Tools, then click Add/Remove. Follow the instructions in the InstallShield wizard to uninstall the applications.

   CAUTION: Be sure to keep a single installation of the previous version of Siebel Tools. You will need this in order to rename repositories when you upgrade the development environment. Otherwise, you will have to reinstall the earlier version of Siebel Tools in order to perform the production upgrade.

2 Install the new Siebel Tools Release 7.x software on all development workstations. For complete instructions on installing Siebel Tools, refer to *Siebel Tools Reference*.

NOTE: If you are uninstalling Siebel eBusiness Applications, Release 6.x.x on the Windows platform, you may experience a failure to uninstall that version due to a bug in InstallShield versions greater than Release 6.0.x.

You will most likely encounter this problem if, on trying to uninstall Siebel eBusiness Applications, you receive a message with the text: `Setup failed to run installation.`

Refer to the chapter on uninstalling Siebel eBusiness Applications in *Siebel Server Installation Guide* for the operating system you are using for a procedure to prevent this error from occurring. You can also refer to Siebel SupportWeb (http://ebusiness.siebel.com/supportweb/).

Upgrading Siebel Gateway and Siebel Servers

Upgrade your Siebel Gateway and Siebel Servers to the Release 7.5 software. If you have multiple Siebel Servers, you will need to perform the upgrade procedure described in this section for all Siebel Servers in your deployment.

Before you proceed, make a copy of the Upgrade Planning Worksheet that you previously completed. You might need to refer to this during the upgrade process for server names and other installation information. You will also need to refer to *Siebel Server Installation Guide* for the operating system you are using for server installation instructions.

CAUTION: Do not attempt to install the new version of Siebel eBusiness Applications software without first removing the previous version as instructed. If you attempt this, your installation might not operate correctly.

Perform the following upgrade procedures on each server:

- Stop all Siebel servers and the Siebel Gateway service.
- Uninstall the previous version of the Siebel Server software.
- Uninstall the earlier version of the Siebel Gateway.
- Install the version 7.5 Siebel Gateway software.
- Install the version 7.5 Siebel Server on all Siebel servers.

Upgrading the Servers on Supported UNIX Platforms

Refer to your previously completed copy of the Upgrade Planning Worksheet for server names and other installation information.

To stop all servers

1. Stop the Siebel Gateway by typing: `stop_ns`
2. Stop all Siebel Servers by typing: `stop_server_ALL`

To uninstall the earlier version of the Siebel Servers

1. If you wish to preserve your installation–specific Siebel Server configuration parameters, record your current configuration.

   You will need to reapply that configuration manually after completing the upgrade. Installation–specific parameters will be lost when you uninstall your existing Siebel Servers. When you install the new Siebel Server Release 7.x software, you can reset these parameters manually through the Server Manager.

2. Uninstall the previous version of the Siebel Server software by deleting the `$SIEBEL_ROOT` directory for the previous version.

   NOTE: To avoid conflicts with the ODBC drivers used by the new Siebel Server software, remove any older versions of the ODBC drivers. If the vendor provided an uninstall option, use it. If the vendor has not provided an uninstall option, contact the vendor for removal instructions.

To uninstall the earlier version of the Siebel Gateway

- Uninstall the Siebel Gateway by deleting your previous installation directory.

  NOTE: If you wish to preserve your previous configuration parameters (for example, if you set `MaxTasks`, `MaxServer`, `MinServers` to specific values) you need to save your previous configuration. Then, after the upgrade, use `srvrmgr` to reset your configuration parameters. This applies to all parameters that have been changed from the standard installation.

To install the version 7.5 Siebel Gateway

- Refer to the appropriate chapter on installing the Siebel Gateway in *Siebel Server Installation Guide* for the operating system you are using.

To install the version 7.5 Siebel Servers

1. Make sure that each application server on which a Siebel Server will be installed has the correct versions of all required third–party software products, as documented in *Siebel System Requirements and Supported Platforms*.

2 Install the new Siebel Server on all application servers as described in *Siebel Server Installation Guide* for the operating system you are using.

NOTE: With the Siebel Enterprise Server (SES) installer, when the Siebel Server is chosen, make sure that you choose the Siebel Database Server as well.

CAUTION: In order to configure and execute Siebel Database Server procedures and maintenance scripts, you must have sufficient access to Siebel Server 7.5 directories:

You must have READ-WRITE access to `bin` directories under Siebel Server executables in the *`SIEBSRVR_ROOT`* directory.

You must have READ-WRITE access to the log directories and upgrade directory.

Installing the Siebel Database Server Software

Install the Release 7.5 Siebel Database Server software onto one Siebel Server. The Siebel Database Server must be installed on a Siebel Server that has already been upgraded to Release 7.5.

NOTE: With the Siebel Enterprise Server (SES) installer, when the Siebel Server is chosen, make sure that you choose the Siebel Database Server as well.

CAUTION: In order to configure and execute Siebel Database Server procedures and maintenance scripts, you must have sufficient access to Siebel Server 7.5 directories:

You must have READ-WRITE access to `bin` directories under Siebel Server executables in the *`SIEBSRVR_ROOT`* directory.

You must have READ-WRITE access to the log directories and upgrade directory.

To install the Release 7.5 Siebel Database Server software, follow the instructions for installing database server components in the *Siebel Server Installation Guide* in the chapter on installing the Siebel Database Server software on your RDBMS. Refer to your completed Upgrade Planning Worksheet for server names and other installation information.

NOTE: Perform the procedure outlined in *Siebel Server Installation Guide* for your platform for installing the stored procedures and user–defined functions.

In order to have Siebel Administrator permissions, you must log in with a valid RDBMS user name and password, and this user name must have “Siebel Administrator” responsibility for the default organization.

The default administrator user name is `SADMIN`. If the user `SADMIN` does not exist or does not have “Siebel Administrator” responsibility for the default organization, you must contact your database administrator to establish this before you proceed with the upgrade.

This attribute can be changed to the login of another “employee” if necessary to meet your business needs.

Installing Additional Languages for a Multilingual Deployment

If your organization deploys internationally and, therefore, requires data to be in multiple languages, you need to install multilingual seed data (for example, lists of views, responsibilities, or system preferences). To do this, you will add new language packs to your database after you have installed the base language for your database server.

- **For upgrades from Release 7.x to 7.5.** Language packs should be installed before you perform a repository merge.
- **For upgrades from Release 6.x to 7.5.** Language packs should be installed after the upgrade of your base language. Adding the language before an upgrade will significantly reduce performance of the repository merge.

See *Siebel Server Installation Guide* for the operating system that you are using for instructions on installing and configuring multiple Siebel language packs and importing a new language to your repository.

NOTE: Unless your database is Unicode-enabled, Siebel Systems does not support installation of any language other than U.S. English on a system based on Japanese, Simplified Chinese, Traditional Chinese, or Korean code pages. No code page restrictions apply to a system supported by a Unicode-enabled database.

About the Siebel Software Configuration Utility

The Database Server Configuration Wizard performs the following tasks:

- Generates your upgrade configuration file (UCF file)
- Validates certain parameters that you enter during an upgrade
- Launches the Siebel Upgrade Wizard

The Upgrade Configuration File

The upgrade configuration file (UCF file) will be used by the Siebel Upgrade Wizard to run the upgrade. This file will be generated by the Database Server Configuration Wizard in $*SIEBSRVR_ROOT/bin*.

The upgrade configuration file uses a standard naming convention:

```
master_upgrade_option_upgrade_type_version.ucf
```

where:

upgrade_option = the type of upgrade you are performing

- ❑ For upgrades of the Siebel Database Schema, the value is `upgrep`.
- ❑ For upgrades of the Custom Database Schema, the value is `upgphys`.

upgrade_type = the environment that you are upgrading

- ❑ For development environment upgrades, the value is `dev`.
- ❑ For production environment upgrades, the value is `prod`.

version = the version from which you are upgrading

For example, the UCF file for an upgrade of the Siebel Database Schema in a development environment from Release 6.0.1 would be named:

```
master_upgrep_dev_601.ucf
```

Parameter Validation

The Database Server Configuration Wizard validates certain parameters, and will not proceed if you enter an invalid parameter. The configuration wizard validates the following information:

- That you have renamed the Siebel Repository to Prior Customer Repository
- That tablespaces exist in your database
- That username and password are valid
- That tableowner and tableowner password are valid
- That the language pack is installed
- That the directories chosen exist

The Siebel Upgrade Wizard

The Database Server Configuration Wizard will automatically launch the Siebel Upgrade Wizard. The Siebel Upgrade Wizard performs the following major tasks:

- Upgrades the Siebel database schema to the new version
- Migrates application data to the new data model structure
- Upgrades Siebel seed data
- Loads new repositories
  - Development environment upgrades load three new repositories:
    - Prior standard repository (version of 5.x or 6.x)
    - New standard repository (version 7.5)
    - New customer repository (version 7.5) which, after the merge, will become the customized 7.5 repository.
  - Production environment upgrades load one new repository:
New Customer Repository (version 7.5).

Upgrading the Siebel Database Schema

The Siebel Upgrade Wizard is restartable at most stages within the upgrade process. For information about restartability, see “Restarting the Upgrade” on page 197.

To configure the Siebel Database Schema on supported UNIX platforms

1. Verify that all servers are stopped:

   - Stop the Siebel Gateway by typing: `stop ns`
   - Stop the Siebel Servers by typing: `stop server ALL`

2. Depending on your shell, source environment variables from *`$SIEBEL_ROOT`* by typing:

   Korn shell

   ```
   . siebenv.sh
   ```

   C shell

   ```
   source siebenv.csh
   ```

3. Set the following environment variables:

   - `SIEBEL_ROOT` should be the path of your Siebel eBusiness Application installation directory.
   - `LANGUAGE` should be set to the language in which the Configuration Wizard prompts appear; for example, `enu` for U.S. English.

   If either of these values is incorrect or empty, reset them using one of the following commands for the shell you are using:

   C shell

   ❏ `setenv LANGUAGE` *`enu (`*where *`enu represents your display language)`*

   ❏ `setenv SIEBEL_ENTERPRISE` *`Enterprise Name`*

   Korn shell

   ❏ `LANGUAGE=enu ; export LANGUAGE`

- ❑ `SIEBEL_ENTERPRISE=`*`Enterprise Name`*` ; export SIEBEL_ENTERPRISE`

4 Navigate to `$`*`SIEBEL_ROOT`*`/bin` and enter:

```
dbsrvr_config.ksh
```

This launches the Database Server Configuration Wizard.

5 Review the values of the following environment variables and confirm whether or not the settings are correct by entering either Y or N.

- ❑ `SIEBEL_ROOT`
- ❑ `LANGUAGE`

NOTE: If either the `SIEBEL_ROOT` or `LANGUAGE` value is not set or is incorrect, you must correct them before proceeding.

---

6 In the Siebel Server Directory screen, specify the path of your Siebel Server root directory, or accept the default by pressing Enter.

7 In the Siebel Database Server Directory screen, specify the path of your database server root directory, or accept the default by pressing Enter.

The RDBMS Platform screen appears.

8 In the RDBMS Platform screen, choose the number that corresponds to the database platform that you are upgrading.

1 - IBM DB2 UDB for Unix and Windows.

2 - Oracle.

The Siebel Database Operation menu appears.

9 From the Siebel Database Operation menu, choose Upgrade Database (`2`).

10 From the Upgrade Options menu, choose Upgrade Siebel Database Schema (upgrep) `(1).`

11 From the Environment Type screen, choose the number that corresponds to the environment that you are upgrading:

```
1 - Development.

2 - Production.
```

Which screen appears next depends on the application that you are upgrading:

- **Siebel eBusiness Applications.** The Current Siebel Version screen appears. Skip Step 12 and continue to Step 13.
- **Siebel Industry Applications or Siebel Financial Services Applications.** The Siebel Industry Vertical screen appears. Continue with Step 12.

12 If you are upgrading a Siebel Industry Application or Siebel Financial Services Application, choose the application from which you are upgrading. Carefully read the descriptions of the available applications. The choices available to you are constrained by the database platform you selected. The following applications may appear:

NOTE: Before choosing your application, see *Siebel System Requirements and Supported Platforms* to verify the supported upgrade paths for your deployment.

- **Siebel Automotive.** Choose this if you are upgrading from Release 6.x of Siebel Automotive.
- **Siebel Communication Media Energy.** Choose this if you are upgrading from Release 5.x or 6.x of Siebel Communications, Media, and Energy.
- **Siebel Consumer Sector.** Choose this if you upgrading from Release 6.x of Siebel Consumer Sector.
- **Siebel Life Sciences.** Choose this if you are upgrading from Release 5.x or 6.x of Siebel Life Sciences.
- **Siebel Industry Solutions.** Choose this if you are upgrading from Release 7.0.3 or 7.0.4 of any of the following applications: Siebel Consumer Sector, Siebel Life Sciences, Siebel Communications, Media, and Energy, Siebel Automotive.
- **Siebel Financial Services.** Choose this if you are upgrading from Release 5.x, 6.x, or 7.x of Siebel Financial Services.

- **Siebel Application.** Choose this if you are upgrading from Release 7.5 of Siebel eBusiness Applications to Release 7.5 of a Siebel Industry Application.
- **Siebel Industry Application.** Choose this if you are upgrading from Release 7.5.1 (limited release) of a Siebel Industry Application to Release 7.5.2 of a Siebel Industry Application.

13 In the Current Siebel Version screen, select the number that corresponds to the version of Siebel you are upgrading from.

The Language Selection screen appears.

14 Enter the number that corresponds to the appropriate language.

15 Enter the ODBC datasource name; for example, `$SIEBEL_ROOT_siebel`.

NOTE: The data source is created automatically by the Siebel Server installation, using the format `$SIEBEL_ROOT_EnterpriseName`. To find the name of your ODBC data source, type: `vi $ODBCINI`.

16 Enter the Database User Name and Password for your database:

a **Database User Name.** User name of the Siebel administrator, for example, `sadmin`.

b **Database Password.** Password for the Siebel administrator, for example, `db2`.

17 Enter the database tableowner name (this is the account that will own the Siebel objects):

Database Tableowner. User name of the tableowner, for example, `siebel`.

Database Tableowner Password. Password for the tableowner, for example, `db2`.

18 Enter the number that corresponds to the platform of your database server operating system:

```
1- Windows

2- Solaris

3- AIX

4- HP-UX
```

19 The prompts that appear next may vary, depending on which database platform you use. See the following table for the prompts that you may receive next. Enter the values that you recorded on your Upgrade Planning Worksheet.

NOTE: Use underscores rather than spaces; these values are case-sensitive.

Field	Value
Index Space	■ DB2 UDB: The name you give to your 4-KB index space for tables ■ Oracle: The name you gave to your index area
4KB Table Space (DB2 UDB only)	■ DB2 UDB: The name you give to your 4-KB table space Note that you must use underscores rather than spaces.
8KB Table Space (DB2 UDB upgrades from 5.x only)	■ DB2 UDB: The name you gave to your 8-KB table space
16KB Table Space (DB2 UDB only)	■ DB2 UDB: 16 KB Table space name
32KB Table Space (DB2 UDB only)	■ DB2 UDB: 32 KB Table space name
Table Space Name	■ Oracle: The name you gave to your data area

After you enter all the requested values, the utility displays the values entered so far.

20 Review the values for accuracy against the values that you recorded in your copy of Appendix A, “Upgrade Planning Worksheet.”

- If the values are correct, launch the Siebel Upgrade Wizard (`srvrupgwiz`) by entering `Y`. The Siebel Upgrade Wizard is explained in "Reviewing the Upgrade Log Files" on page 153.

  NOTE: If a program or system error occurs and you need to rerun the Siebel Upgrade Wizard, you can do so, starting at the point at which the wizard failed, by navigating to `$SIEBEL_ROOT/bin` and typing:

  ```
  srvrupgwiz /m master_upgrade_option_upgrade_type_
  version.ucf
  ```

  for example, `srvrupgwiz /m master_upgrep_dev_601.ucf`

- If you need to change the values you previously entered, rerun the Configuration Wizard by entering `dbsrvr_config.ksh`. Rerunning the Configuration Wizard will allow you to reconfigure with different values.

The Siebel Upgrade Wizard displays a message when the upgrade is complete.

CAUTION: The upgrade process executes in a UNIX shell. Do not click inside this shell while the upgrade is running. If you accidentally click inside the UNIX shell, the process will stop executing and you will need to restart the upgrade.

Reviewing the Upgrade Log Files

After an upgrade, depending on the trace level that you set, you need to review several log files generated. Review these log files for unacceptable errors.

- If the upgrade completed successfully, there are several log files that you can safely ignore:
  - `siebel.log, siebel_01.log, siebel_02.log, siebel_03.log...`
  - `srvrupgwiz_01.log, srvrupgwiz_02.log, srvrupgwiz_03.log...`

    NOTE: Pay attention to log file names, because there are similarly named log files that you do need to review: `srvrupgwiz1.log`, `srvrupgwiz2.log`, and subsequent log files.

  - Any other log file that existed prior to the start of the upgrade
- If you encountered an error during the upgrade, review the log files generated by the upgrade wizard, such as `srvrupgwiz1.log`, `srvrupgwiz2.log` within the *`SIEBEL_ROOT`*`/log` directory. (The name of the log file will increment for subsequent log files that are created when the upgrade wizard encounters a problem and the user attempts a retry.)

  Review the end of the `srvrupgwiz` log file for details about the latest failure. If the step that failed was not a native SQL step (which would be listed in the `srvrupgwiz` log file), then it occurred as part of an external utility, for which you need to review a corresponding log file, identified by the /L parameter.

  NOTE: Archive your log files in case future analysis is required by technical support. (Preserve the date and time of the log files, because a copy of the files will overwrite the date and time stamp.)

The log files may include errors that are expected and benign. Compare any error messages found in the log files to the sample error messages in the `errors.rtf` file located in the *`DBSRVR_ROOT\PLATFORM`* directory.

To review the log files for unacceptable errors

1 Print the `errors.rtf` file. This file is located in the installation subdirectory for your database platform.

2 Sort the files in the *$SIEBEL_ROOT*/`log` subdirectory according to date. Under both Solaris and AIX, sort the files by entering the command: `ls -ltr`

- Only one of each type of error occurring in a particular log file appears in the `errors.rtf` file.

  NOTE: The log files generated by the repository upgrade wizard (for example, `srvrupgwiz1.log`) appear in `errors.rtf` as `upgwiz1.log`, `upgwiz2.log` (which increment for additional log files).

- If a log file is not listed in the `errors.rtf` file, then there are no acceptable error messages for that log file.

3 Open each log file, starting with the earliest, and search for errors. (Errors are either tagged with the word “error” or enclosed in square brackets “[...].”)

NOTE: It is important that you start with the earliest log file. It will shorten your research time if you find unacceptable errors in an early log file.

4 For each error found, look for that error in the list of acceptable errors documented in the `errors.rtf` file.

- If you locate the error in the `errors.rtf` file, then it is acceptable and no action is required. Continue to review the errors found in the log file.

  NOTE: Certain acceptable errors may appear multiple times in a log file. This is acceptable. No action is required.

- If you find an error that is not listed in the `errors.rtf` file, then it is unacceptable. You must correct the condition that caused the error before you rerun the Upgrade Wizard.

  Report the error to Siebel Technical Support. *Do not proceed with the upgrade.*

5 Repeat the previous step for each log file.

Do not review error numbers only, because these may have changed following installation of a new driver version. Instead, compare the actual error descriptions to find out which are acceptable errors for this platform.

Although other errors are rarely encountered, this review is critical. Certain errors, such as a failure to create indexes, may result in performance problems or anomalous behavior in Siebel eBusiness Applications.

NOTE: Do not proceed with the upgrade until unacceptable errors have been corrected. If you cannot correct the error, contact Siebel Technical Support or Professional Services to report the error in detail.

Troubleshooting the Siebel Database Schema Upgrade

Typical problems that might occur during the upgrade of your Siebel Database Schema (upgrep) could result from a lack of storage space or insufficient user privileges.

Recovering from a Failed Siebel Database Schema Upgrade

If the repository upgrade fails due to insufficient space allocated on the database, you must complete the following procedures.

To recover from a failed Siebel database schema upgrade

1. Back up your complete set of log files, from the beginning of the upgrade process to the point at which it stopped, to another directory.
2. Read the `srvrupgwiz1.log` file and associated log files to determine the failure. See “Reviewing the Upgrade Log Files” on page 153.
3. Take the necessary corrective action. Depending on the errors that you find, you may need to ask your DBA to extend the database.
4. Resume the upgrade wizard. It will continue from the point at which it failed.

NOTE: To resume the upgrade wizard, navigate to `$SIEBEL_ROOT/bin` and type:

```
srvrupgwiz /m master_upgrade_option_upgrade_type_
version.ucf
```

For example, to restart a development environment upgrade from Release 6.0.1 to Release 7.x, type:

```
srvrupgwiz /m master_upgrep_dev_601.ucf
```

Restarting the Upgrade

The Siebel Upgrade Wizard is restartable at most stages within the upgrade process. If the Siebel Upgrade Wizard encounters an error during the upgrade process, it will stop at that point. If your upgrade stops due to an error, perform the following steps.

To restart the upgrade

1. Carefully review the relevant log files to make sure that your upgrade has completed successfully up to that point.
2. Before you restart the upgrade (after any break in the upgrade process) back up your complete set of log files, from the beginning of the process to the point at which it stopped, to another directory. This will maintain a complete record of your log files, and prevent your previous log files from being overwritten, which could prevent accurate diagnosis of the reason for the break in the upgrade.
3. After you resolve the failure, you may restart the upgrade. The upgrade will continue from the last step that completed successfully.

CAUTION: If problems with your environment prevent the upgrade from restarting, you need to restore the database from the prior base version (the version which you are upgrading from). For example, environment problems may occur when table creation fails due to a database problem (insufficient storage or network problems), which cause subsequent upgrade steps to fail. If you need to restore your database and restart the upgrade, you should delete or store the upgrade log files that were generated in the SiebServ\Log directory to a zip file. You should also delete the log file, `state.log`, from the *`SIEBSRVR_ROOT`*`/upgrade/driver_upgrep_dev_version` directory.

Launching the Siebel Upgrade Wizard

The Siebel Upgrade Wizard launches from the final prompt of the Database Server Configuration Utility. However, if a program or system error occurs and you need to rerun the Siebel Upgrade Wizard, you can do so, starting at the point at which the wizard failed, by using the following procedure.

To relaunch the Siebel Upgrade Wizard or to restart the upgrade

1 Navigate to $*SIEBEL_ROOT*/bin and type the following command:

```
srvrupgwiz /m master_upgrade_option_upgrade_type_version.ucf
```

For example, to restart a development environment upgrade from Release 6.0.1 to Release 7.x, type the following command:

```
srvrupgwiz /m master_upgrep_dev_601.ucf.
```

2 To begin the upgrade of your repository, at the prompt, press Enter.

The Siebel Upgrade Wizard will notify you when the upgrade process is complete.

You have finished upgrading your repository and are ready to review the log files for errors. See “Reviewing the Upgrade Log Files” on page 153.

Making a Backup of the Database Repository

Back up your database repository after a successful upgrade of the Siebel Database Schema.

NOTE: This backup will enable you to restore your pre-merge database if you need to recover from a failed repository merge. See “Recovering from a Failed Merge” on page 187.

Adding New License Keys

CAUTION: For production environment upgrades, skip this procedure. You will perform this step after upgrade of the *custom database schema*.

With the new Release of Siebel eBusiness Applications, you received one or more license keys that must be added to the development database. You must add all of the new license keys to enable the new release of the Siebel eBusiness Applications, which you will use in the next steps of the upgrade process.

To add new license keys

1. Start Siebel Tools version 7.x from a development workstation and log on to the database server as the Siebel administrator.
2. Add your new license keys.

Preparing the Prior Customer Repository for the Merge

CAUTION: The repository merge procedures only apply to development environment upgrades. If you are performing a production environment upgrade, skip to "Upgrading the Custom Database Schema" on page 189.

You will perform several procedures to prepare the Prior Customer Repository for the repository merge.

- **Run the Repository Preparation Wizard.** The Repository Preparation Wizard needs to be run on the Prior Customer Repository before you proceed with the repository merge. This utility is invoked from Siebel Tools and will be used to perform the following procedures:
  - **Migrate Strings**
  - **Merge Labels and Fields**
  - **Merge Templates**

  See "Migrate Strings, Merge Labels and Fields, and Merge Templates."

CAUTION: If you are upgrading from Release 7.0.x to Release 7.5, do not run the Repository Preparation Wizard. You already completed this procedure during your upgrade to Release 7.0.x.

- **Upgrade Copied Objects.** The ability to upgrade copied objects is a new feature for Release 7.x upgrades. Copied objects are the customized objects from prior versions of Siebel software. In prior upgrades, to Release 6.x, only Siebel objects were upgraded. In Release 7.x upgrades, your custom objects will be upgraded if they have an upgrade ancestor specified. See "Automatic Upgrade of Copied Objects" on page 165.

Migrate Strings, Merge Labels and Fields, and Merge Templates

CAUTION: The repository merge procedures only apply to development environment upgrades. If you are performing a production environment upgrade, skip to "Upgrading the Custom Database Schema" on page 189.

Before you run the repository merge, you need to run the Repository Preparation Wizard from Siebel Tools to perform the following operations:

- **Migrate strings.** Release 7.x supports locale strings in object-specific tables. In order to move the S_MSG data in your environment to the new locale table structure in Release 7.x, you need to run the Migrate Strings utility.

  NOTE: Migrate strings for each language that was supported in your prior repository.

- **Merge labels and fields.** Release 7.x merges labels with controls so that the label is now recognized as the caption property of the control. Before you run the repository merge, you need to run the Repository Preparation Wizard to merge labels, controls and fields from the previous version based upon form applet layout.

  NOTE: If you are upgrading to Siebel Financial Services 7.x, you will not use the default input file to merge labels and fields. See the Caution note in Step 3 on page 162.

- **Merge Web templates.** Release 7.x merges web templates in Siebel Tools to only use Base and Edit/Query/New for the majority of the out of the box applets. Previous releases of Siebel used four separate templates to deliver the same functionality. Before you run the repository merge, you need to run the Merge Templates utility to merge your applet web templates.

The Repository Preparation Wizard will prompt you to confirm that you completed the following steps before it will proceed with the repository merge.

NOTE: If your upgrade fails while you are preparing the prior customer repository for merge, you need to restore your database to the most recent backup and complete repository preparation procedures. For example, if your upgrade fails after you successfully completed the "Merge Labels and Fields" part of repository preparation, you would restore your database to the point after which you merged labels and fields.

To migrate strings, merge labels and fields, and merge applet Web templates

1. In Siebel Tools, navigate to the Tools menu, and then choose Tools > Upgrade > Prepare Repository.

   Choose Prior Customer Repository as the repository that you wish to prepare.

   The String Migration window appears.

2. In the String Migration window:

   a. Select the language for which you wish to migrate strings.

   b. To log migrated strings, click in the check box beside `Log migrated strings`, then click the browse button to specify the log file.

      To continue, click Next. The Merge Labels and Fields window appears.

3. In the Merge Labels and Fields window:

a Specify the input file by clicking the browse button.

NOTE: The utility will operate on every form applet except those specified in the input file. The default input file is `applets.txt`. Modify the input file only if you have additional applets that should not be merged (those developed for Handheld or Wireless applications).

CAUTION: If you are upgrading to Siebel Financial Services 7.x, you should not use the default input file, applets.txt, because the repository preparation utility will operate on every form applet except those specified by the input file, and Siebel Financial Services applications contain additional applets that should not be merged. Instead, you should browse to `SIEBEL_ROOT\Tools\bin` directory and select the input file named `fins_applets.txt`.

b Specify the location of Web templates.

To continue, click Next. The Merge Applet Web Templates window appears.

4 In the Merge Applet Web Templates window, click the browse button to specify the input file, then click Next.

NOTE: The utility will operate on every form applet except those specified in the input file. The default input file is `applets.txt`. Modify the input file only if you you have additional applets that should not be merged (those developed for Handheld or Wireless applications).

The Merge Applet Web Templates window appears. Click OK to confirm that you want to proceed.

The wizard will prepare your Prior Customer Repository for the merge.

The Repository Preparation Wizard records the results of the labels and fields merge and the template merge into the `labelmerge.txt` file and the `templatemerge.txt` file, respectively. If you want to view information about how the Repository Preparation wizard prepared your repository for merge, you can check these files, located in the *`Siebel_Tools`*`\temp` directory. More information about these log files is provided in the table below.

File Name	Description
labelmerge.txt	Generated by label and field merge. Lists all applets that are being modified. For each applet modified, displays whether the label and field merge was successful. If a control already has a caption, the old caption is logged for that control.
templatemerge.txt	Generated by the template merge. Lists all applets that are being modified and displays whether the merge was successful. For each applet, displays the following: ■ The template being used as the source (other templates are merged into this template.) ■ The Applet Web Templates being inactivated. ■ Controls whose types are changed to ensure that they only appear in the correct modes.

Automatic Upgrade of Copied Objects

CAUTION: The repository merge procedures only apply to development environment upgrades. If you are performing a production environment upgrade, skip to "Upgrading the Custom Database Schema" on page 189.

Siebel Tools allows copied objects to inherit some of the behavior of their ancestors, which makes it easier to upgrade Siebel applications, reduces the time and cost of adjusting an application after an upgrade, and also supports parallel development by allowing some frequently used objects to be copied.

Certain repository objects that are copied during configuration can be upgraded with a new property called Upgrade Ancestor that stores the name of the ancestor object. This allows copied objects to be upgraded in the same way as the ancestor objects from which they were copied. Thus when you copy an existing object, you can specify its upgrade ancestor; during an upgrade the copied object will be upgraded the same way as the original. This feature is available only for objects of type Applet, Business Component, Report, and Integration Object.

NOTE: Use of the Upgrade Inheritance feature will slow the performance of the repository merge.

Upgrade Inheritance functionality:

- The Upgrade Ancestor property stores the name of the ancestor object (that is, the one from which the current object was copied).
- If the Upgrade Ancestor property is not null, you can upgrade the copied object as if it were the ancestor object.
- No special action is taken during import even if the "Upgrade Ancestor" property is specified, because this property is specific to Application upgrades. However, imported objects can have the Upgrade Ancestor property set. When the next application upgrade is done, the property is taken into account.
- Inheritance does not apply to patch files. The "Upgrade Inheritance" property is applicable only during application upgrades. It is not taken into account during the application of a patch and no action is performed based on this property.

- During the merge, the newly created objects are given all the changes corresponding to its ancestor. Objects with the Upgrade Ancestor property include:
    - Applets
    - Business Components
    - Integration Objects
    - Reports

  Basically, you can create a copy of an existing object (applets, business components, integration objects, and reports) and specify an Upgrade Ancestor.

Upgrade Inheritance Scenario

For example, you may want to make a copy of the Account List Applet and call it the Premium Account List Applet. This new applet may differ from the original one in that it has a special search specification that is displayed only in those accounts that are considered premium accounts. In a subsequent release, Siebel eBusiness Applications may include additional standard list columns to the Account List Applet. During an application upgrade, your Account List applet and the Premium Account List Applet will retain the configuration changes you made. However, both applets will receive the new standard list columns added in the new version because of Upgrade Inheritance functionality. Without this new feature, the copied applet would not receive the new list columns during the upgrade process.

How Enhancements Are Applied During an Upgrade

During upgrades, it is very common that objects in the repository are changed. For example, an applet might have a few list columns added or a business component might have some fields and a multi-value link added. To do this, the objects that need to be changed during the upgrade are recognized by their Name property. For example, you would query the repository for the Account BC and add the necessary new items to it. If you did not have the Upgrade Inheritance feature and the Account BC had been copied as Acme Account, you would not recognize the new BC as a copy of the Account BC and would not add the required changes to the copy during the upgrade. These additions might be minor, but often these omissions can cause numerous application errors after the upgrade and can be time consuming to detect and correct.

During an upgrade, the Upgrade Inheritance feature makes sure that copied objects receive the same changes that are applied to the object from which they were copied. This is done automatically by the upgrade utility, and there is no manual step involved except for specifying the property.

NOTE: This functionality is applied only to the following object types: business component, applet, integration object, and report.

Choosing an Upgrade Ancestor

When choosing an upgrade ancestor for an object, the picklist of objects displayed from which you can choose varies depending on the object type. The picklist has the following constraints for each object type:

- Applets that have the following characteristics:
  - Table is the same as the current applet buscomp
  - Class is the same as the current applet class
  - Upgrade Ancestor is null
  - Available out of the box—Siebel supplied object: shipped as part of the Standard Siebel repository
- Reports that have the following characteristics:
  - Buscomp is the same as the current report buscomp
  - Class is the same as the current report class
  - Upgrade Ancestor is null
  - Available out of the box; Siebel supplied object
- Business Components that have the following characteristics:
  - Bus Comp is the same as the current business component
  - Class is the same as the current business component
  - Upgrade Ancestor is null

  - Available out of the box; Siebel supplied object

- Integration Objects that have the following characteristics:
  - Base Object Type is the same as the current Base Object Type
  - Business Object is the same as the current business object
  - Upgrade Ancestor is null
  - Available out of the box; Siebel supplied object

The constraint requires that these picklists show only standard objects; this can be relaxed by setting a flag found in View > Options > General. This may be appropriate for customers that will use the inheritance feature to support distributed development. Relaxing this constraint does not change the fact that ancestor objects must be found in the New Standard repository in order to be applied to their descendants during a merge.

NOTE: If an object that does not exist in the 7.5 New Standard repository is specified as an ancestor object, you will receive error messages during the repository merge process in the merge.txt file. These errors are acceptable. However, you may want to manually update the descendant objects of the ancestor object because these objects were not updated with the characteristics of the ancestor object during the merge.

Repository Location of the Upgrade Ancestor

During the application upgrade, the contents of three repositories are compared to produce the final, postupgrade repository which contains both the customizations made by the customer as well as any enhancements added in this release during the upgrade. The three repositories compared are the following:

- Prior Standard Repository: Ancestor Repository.
- Prior Customer Repository: Ancestor Repository that has been customized by the client.
- New Standard Repository: New Siebel Repository.

The Upgrade Ancestor object of a copied object must exist in the New Standard repository in order for any enhancements to be applied to descendants during the merge.

- New Customer Repository: New Siebel Repository customized by the client. Before the merge, this is the New Siebel Repository; after the merge, it is a Release 7.5 repository which contains customizations made by the customer in the previous release, identified in the Prior Customer Repository.

The outcome of the upgrade ancestor only affects the New Customer Repository.

Configuration Steps for Upgrade Inheritance

CAUTION: The repository merge procedures only apply to development environment upgrades. If you are performing a production environment upgrade, skip to "Upgrading the Custom Database Schema" on page 189.

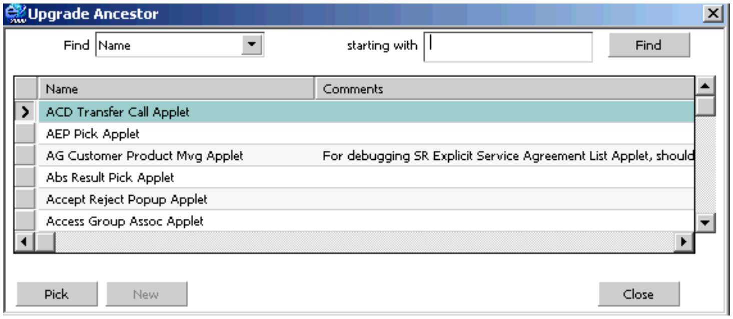
After you have decided it is necessary to create a copy of an object in the repository, it is easy to specify the parent object from which the copy was created. After creating the copied object, specify the parent object name in the Upgrade Ancestor property of the copied object. This is what allows the copied object to be recognized as a copy during the application upgrade, and it is changed along with its parent object.

NOTE: You must manually populate this property since it is not automatically populated for you when you copy an object. Remember that this property can only be populated if the copied object is an applet, business component, integration object, or report object type.

To copy an applet, business component, integration object, or report object and children

1. Select the Object type (Applet, Business Component, Integration Object, or Report) in the Object Explorer.
2. Select an entry in the Object list applet.
3. Choose Edit > Copy Record to create a copy of this record.
4. In the new record, fill in a new name in the Name field.

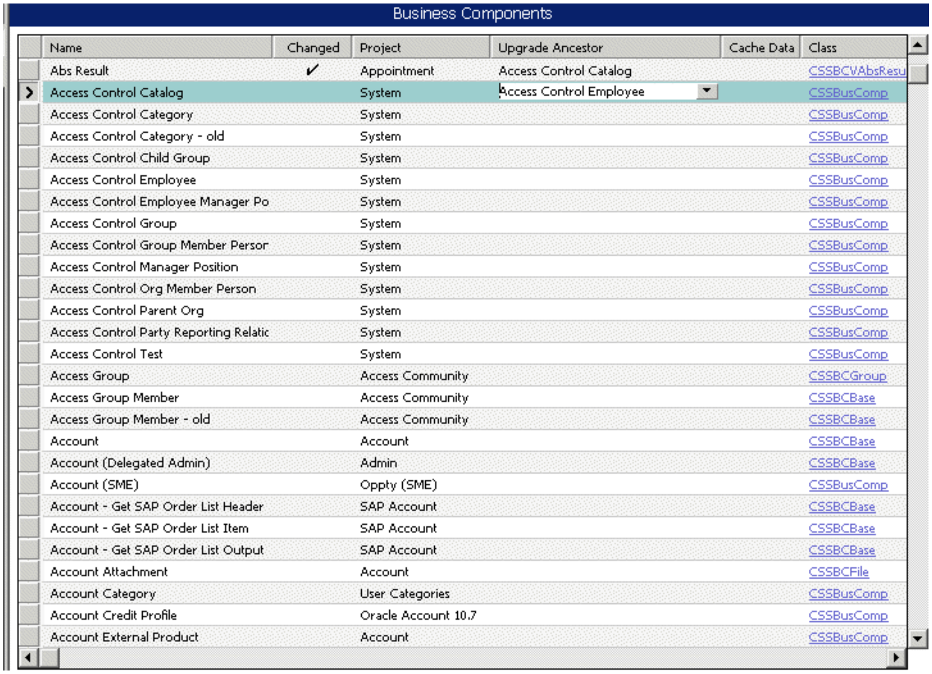
5 In the Upgrade Ancestor field, select a value from the Upgrade Ancestor dialog box.



Name	Comments
ACD Transfer Call Applet	
AEP Pick Applet	
AG Customer Product Mvg Applet	For debugging SR Explicit Service Agreement List Applet, should
Abs Result Pick Applet	
Accept Reject Popup Applet	
Access Group Assoc Applet	

The picklist shows all the other business components, applets, integration objects, and reports in the repository.

The following figure shows that the Access Control Employee BC is the Upgrade Ancestor for the copied BC.



Business Components

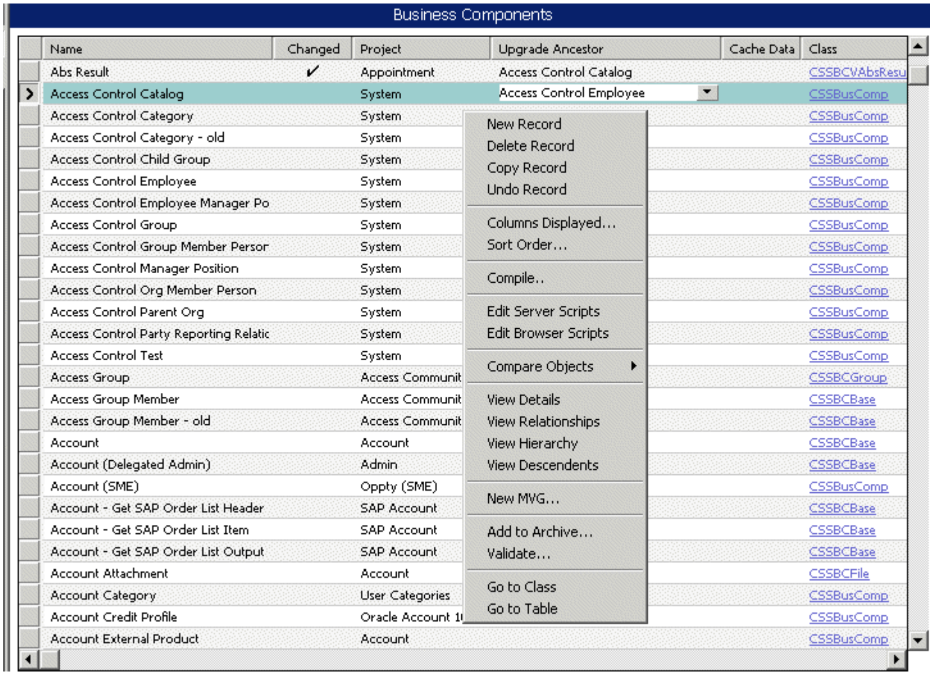
Name	Changed	Project	Upgrade Ancestor	Cache Data	Class
Abs Result	✓	Appointment	Access Control Catalog		CSSBCVAbsResu
Access Control Catalog		System	Access Control Employee		CSSBusComp
Access Control Category		System			CSSBusComp
Access Control Category - old		System			CSSBusComp
Access Control Child Group		System			CSSBusComp
Access Control Employee		System			CSSBusComp
Access Control Employee Manager Po		System			CSSBusComp
Access Control Group		System			CSSBusComp
Access Control Group Member Persor		System			CSSBusComp
Access Control Manager Position		System			CSSBusComp
Access Control Org Member Person		System			CSSBusComp
Access Control Parent Org		System			CSSBusComp
Access Control Party Reporting Relatic		System			CSSBusComp
Access Control Test		System			CSSBusComp
Access Group		Access Community			CSSBCGroup
Access Group Member		Access Community			CSSBCBase
Access Group Member - old		Access Community			CSSBCBase
Account		Account			CSSBCBase
Account (Delegated Admin)		Admin			CSSBCBase
Account (SME)		Oppty (SME)			CSSBusComp
Account - Get SAP Order List Header		SAP Account			CSSBCBase
Account - Get SAP Order List Item		SAP Account			CSSBCBase
Account - Get SAP Order List Output		SAP Account			CSSBCBase
Account Attachment		Account			CSSBCFile
Account Category		User Categories			CSSBusComp
Account Credit Profile		Oracle Account 10.7			CSSBusComp
Account External Product		Account			CSSBusComp

To view all descendants or copies of an object

1 Right-click on an object.

A dialog box appears.

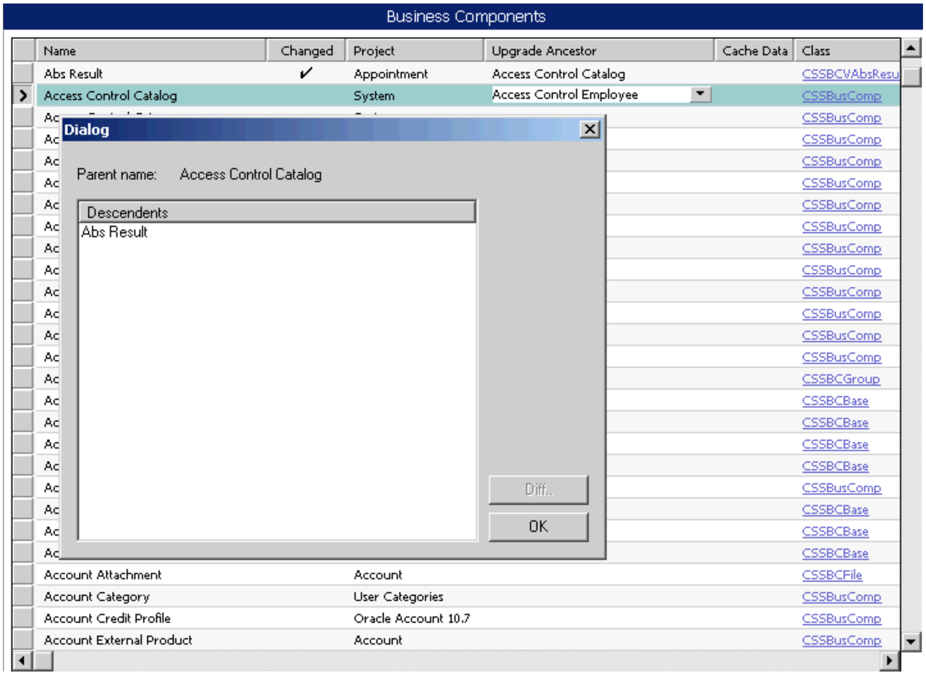
2 Select View Descendants from the dialog box.



Business Components

Name	Changed	Project	Upgrade Ancestor	Cache Data	Class
Abs Result	✓	Appointment	Access Control Catalog		CSSBCVAbsResu
Access Control Catalog		System	Access Control Employee		CSSBusComp
Access Control Category		System			CSSBusComp
Access Control Category - old		System			CSSBusComp
Access Control Child Group		System			CSSBusComp
Access Control Employee		System			CSSBusComp
Access Control Employee Manager Po		System			CSSBusComp
Access Control Group		System			CSSBusComp
Access Control Group Member Persor		System			CSSBusComp
Access Control Manager Position		System			CSSBusComp
Access Control Org Member Person		System			CSSBusComp
Access Control Parent Org		System			CSSBusComp
Access Control Party Reporting Relatic		System			CSSBusComp
Access Control Test		System			CSSBusComp
Access Group		Access Communit			CSSBCGroup
Access Group Member		Access Communit			CSSBCBase
Access Group Member - old		Access Communit			CSSBCBase
Account		Account			CSSBCBase
Account (Delegated Admin)		Admin			CSSBCBase
Account (SME)		Oppty (SME)			CSSBusComp
Account - Get SAP Order List Header		SAP Account			CSSBCBase
Account - Get SAP Order List Item		SAP Account			CSSBCBase
Account - Get SAP Order List Output		SAP Account			CSSBCBase
Account Attachment		Account			CSSBCFile
Account Category		User Categories			CSSBusComp
Account Credit Profile		Oracle Account 1			CSSBusComp
Account External Product		Account			CSSBusComp

The following dialog box appears showing the Parent Name and Descendants.



Business Components

Name	Changed	Project	Upgrade Ancestor	Cache Data	Class
Abs Result	✓	Appointment	Access Control Catalog		CSSBCVAbsResu
Access Control Catalog		System	Access Control Employee		CSSBusComp
Ac					CSSBusComp
Ac					CSSBusComp
Ac					CSSBusComp
Ac					CSSBusComp
Ac					CSSBusComp
Ac					CSSBusComp
Ac					CSSBusComp
Ac					CSSBusComp
Ac					CSSBusComp
Ac					CSSBusComp
Ac					CSSBusComp
Ac					CSSBusComp
Ac					CSSBCGroup
Ac					CSSBCBase
Ac					CSSBCBase
Ac					CSSBCBase
Ac					CSSBCBase
Ac					CSSBusComp
Ac					CSSBCBase
Ac					CSSBCBase
Ac					CSSBCBase
Account Attachment		Account			CSSBCFile
Account Category		User Categories			CSSBusComp
Account Credit Profile		Oracle Account 10.7			CSSBusComp
Account External Product		Account			CSSBusComp

Dialog

Parent name: Access Control Catalog

Descendents
Abs Result

UI objects should be copied if the look-and-feel of the application will change significantly or if there is a difference needed between two objects (that is, different search specifications on two applets). Business Components should only be copied after all other configuration approaches have been exhausted, and copying is clearly the only solution. The issues involving repository maintenance and specialized classes still are present with copied BCs. Upgrade Inheritance functionality allows certain copied objects to be upgraded and inherit the same characteristics that the parent object possessed. This avoids postupgrade errors and configuration problems.

Propagating Changes from a Parent to Descendants

It is possible to propagate changes that are made to the parent to the descendants of that parent. This is accomplished with the new Object Comparison and Synchronization feature, which allows two objects to be compared, and the differences between the two objects to be applied to one another to keep them synchronized.

NOTE: The Object Comparison and Synchronization feature can be used to compare any two objects. The objects do not need to have a parent-descendant relationship.

CAUTION: All objects relating to the Team Forecast View (SWG) and the Personal Forecast View (SWG) must not be renamed because they do not exist in the Release 7.x repository without the “(SWG)” suffix.

Performing a Repository Merge

CAUTION: Repository merge procedures only apply to development environment upgrades. If you are performing a production environment upgrade, skip to "Upgrading the Custom Database Schema" on page 189.

During the repository merge, objects from the Prior Siebel Repository, Prior Customer Repository, and New Siebel Repository are compared by name to identify the total set of object differences. The process also determines how conflicts between repository changes will be resolved as they are merged into the New Customer Repository.

There are three basic categories of object differences:

- New
- Deleted
- Modified

The repository merge executes the following processing steps to identify object differences:

- **New or deleted objects.** Identify objects that the customer has added by comparing their names in the Prior Customer Repository with the Prior Siebel Repository.

  All new customer objects are carried over from the Prior Customer Repository to the New Customer Repository. The repository merge typically avoids deletion of objects. Most of the objects that are deleted in the Prior Customer Repository will reappear after the merge. The merge does this to avoid accidental deletion of objects which may be required. It does, however, allow deletion of specific types of objects. Such objects will be deleted from the New Customer Repository during the merge.

  Objects of the following types will be deleted from the New Customer Repository:

  - Control
  - List Column
  - Page Tab
  - Chart
  - Applet Web Template Item
  - View Web Template Item

- **Objects with altered attributes.** Identifies objects that exist in both the Prior Customer Repository and the New Siebel Repository, and compares the attributes of each object to determine if they have been modified. Attribute comparisons are of interest only for those attributes which were changed by the customer.

  If an object attribute was altered in the Prior Customer Repository, but not in the New Siebel Repository, the customer's attribute value will be merged into the New Customer Repository.

A conflict occurs, however, if an object attribute was altered in both the Prior Customer Repository and the New Siebel Repository, in which case the values in all three repositories would be different. In this event, the repository merge process uses the setting of the object attribute's `StandardWins` flag to determine how to resolve the conflict. If this is set to `Y`, the attribute value from the New Siebel Repository will be used; if this is set to `N`, the attribute value from the Prior Customer Repository is used. Conflict resolutions can be overridden for each object attribute in the New Customer Repository. For examples of conflict resolution, see "Examples of Conflict Resolution" on page 185.

Improving Performance of the Merge

There are several ways in which you can reduce the time required to complete the merge.

1. Optimize the machine on which you are running the merge.

   - Use a workstation with a minimum of 512 megabytes (MB) of RAM.
   - Allocate at least 2 GB of virtual memory, and a 2 GB page file. If the amount of virtual memory on the system is too low, performance will degrade significantly.
   - Close all other applications.
   - Close all services.
   - Defragment the disk. Fragmentation significantly affects system performance.
   - On the workstation, check that the environment variable `SIEBEL_LOG_EVENTS` is set to zero. To check, enter the following command at the MS DOS prompt: `echo %SIEBEL_LOG_EVENTS%` If this variable is not set, no action is required. However, if `SIEBEL_LOG_EVENTS` is returned with a value other than zero, you must set it to zero by performing the following steps:

     ❑ Close Siebel Tools and any other Siebel client applications.

- ❏ Navigate to Start > Settings > Control Panel > System > Environment.
- ❏ In the Environment dialog box, in the System Variables box, select `SIEBEL_LOG_EVENTS`. Enter 0 in the Value box, and click Set. Click OK.
- ❏ Relaunch Siebel Tools. The new setting becomes active.

NOTE: The steps you need to take to set this variable may vary depending on the operating system you are using.

2 Optimize your database, since database performance can cause the merge to slow down considerably.

- ■ Make sure the database has sufficient rollback segments. For more information about rollback segments, see *Siebel Server Installation Guide* for your operating system.

NOTE: If your RDBMS is Oracle, verify that you have only one large rollback segment on line that is appropriately sized so that the largest of transactions can be accommodated. Take all other rollback segments off line.

- ■ Make sure that temporary tablespace has enough space allocated
- ■ Make sure the database has enough space allocated.
- ■ Make sure that the top-most logging applet in tools has no extra rows (from previous merge runs) when starting the merge.
- ■ Make sure that the database is not loaded with users when merge is run (no other users should be connected).
- ■ Delete extra repositories from the database using Siebel Tools. Running the merge on a database with more than the four (4) repositories which are needed for the merge will degrade merge performance. Before deleting extra repositories, make backups.

NOTE: Deletion of extra repositories may take a few hours.

3 Run the merge on a different machine than the database server. When you run the merge on a different machine, you can increase the foreground performance of the merge. To do this, increase the application performance setting on the machine and ensure that the Siebel Tools application is the foreground application.

To increase the foreground performance of the merge

1 Navigate to Start > Control Panel > System.

2 Select the Advanced tab.

3 Select the Performance Options button.

4 In the Application Response box, click the Applications radio button and click OK.

5 While the merge process is running, click on the title bar of the Siebel Tools application to ensure that the Siebel Tools application is the foreground application on the machine.

NOTE: After the merge process has finished, set the Performance setting back to its former value.

Performing the Repository Merge

The configuration utility that you ran in the section "Upgrading the Siebel Database Schema" loaded two version 7.5 standard repositories. You will now use Siebel Tools to merge your existing custom configuration into one of these new repositories, creating a version 7.5 custom configuration that includes all of your previous configuration changes.

The four repositories that currently exist in your development database are listed in Table 14.

Table 14. Development Database Repositories

Repository Name	Description
Prior version 5.x, 6.x, or 7.0.x Siebel Repository	Standard version 5.x, 6.x, or 7.0.x repository, depending on the version from which you are upgrading.
Prior Customer Repository	Customized version 5.x, 6.x, or 7.0.x repository, depending on the version from which you are upgrading.

Table 14. Development Database Repositories

Repository Name	Description
New Siebel Repository	Newly loaded version 7.5 standard repository.
New Customer Repository	Newly loaded version 7.5 repository into which your custom configuration will be merged.

Follow the guidelines provided in “Improving Performance of the Merge” on page 176 to optimize performance of the repository merge.

- The merge is a memory–intensive process and will fail if insufficient memory is available on the Siebel Tools workstation. Before beginning a repository merge, make sure that the following preparations have been completed on the developer workstation. Make sure that the developer workstation on which Siebel Tools is running has been upgraded to the newest available version.

- Use a workstation with a minimum of 512 megabytes (MB) of RAM and at least 2–GB of virtual memory or a 2–GB page file. Otherwise, the merge will not complete successfully.

- Close all other applications before performing a repository merge.

  NOTE: Some repositories may require additional memory, based on level of customizations.

- If necessary, increase the swap space, using the Control Panel System applet, and then restart the development workstation before proceeding.

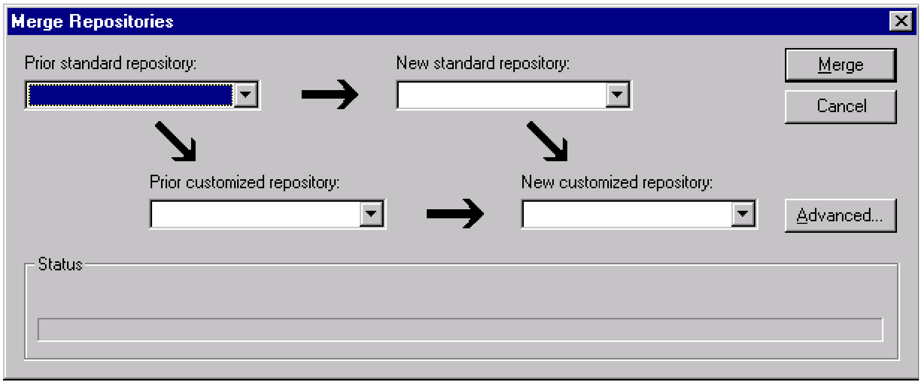
To merge the repository

1. Log on to a Siebel Tools client and use the File > Open Repository command to open the Prior Customer Repository.

2 Choose Tools > Upgrade > Upgrade Application.

CAUTION: Be sure to open the Prior Customer Repository, not another repository. *Later steps in the repository merge process will fail if you have the wrong repository open.*

The Merge Repositories dialog box appears.


Merge Repositories
Prior standard repository:
New standard repository:
Merge
Cancel
Prior customized repository:
New customized repository:
Advanced...
Status

3 In the Merge Repositories dialog box, choose the appropriate repository name from each picklist, using the repository names listed in the following table.

Drop–Down List Item	Value to Choose
Prior Standard Repository	Prior 5.x Siebel Repository, prior 6.x Siebel Repository or prior 7.0.x Siebel Repository, as appropriate for the version from which you are upgrading
Prior Customized Repository	Prior Customer Repository
New Standard Repository	New Siebel Repository
New Customized Repository	New Customer Repository

4 Review the settings in the Merge Repositories dialog box, and then click Merge.

The repository merge process may take, on average, five to seven hours to complete. Timings may vary greatly depending on the kind of machine, the hardware configuration, virtual memory allocation, the use of the upgrade inheritance feature, and level of customizations in the customer repository (such as new records or changed attributes). In addition to merging the base repository, all locales are merged. Additional time should be planned for each language, including the base language.

Customizations are moved to the New Customer Repository, which results in a large number of database operations (inserts and updates). For each of these operations, logging records are created, and these log records also affect performance. If the repository is large, or the database setup is not optimal, this may take much longer.

Reviewing the Merge Results

This section provides an overview of the repository merge process and describes how to review and modify, if desired, the results of the repository merge.

During the merge, errors are reported in the status view and recorded in the merge.txt file. There are no acceptable errors for the repository merge. The only exception to this rule is errors that occur as a result of the upgrade ancestor feature. These errors are acceptable. For more information, see “Automatic Upgrade of Copied Objects” on page 165. After the merge has completed, you must review the results and resolve any errors and undesired conflicts before you proceed with the remainder of the upgrade.

NOTE: If an object that does not exist in the 7.x New Siebel repository is specified as an ancestor object, you will receive error messages in the merge.txt file. These errors are acceptable. An example of an error of this type is: `!!ERROR::CANNOT upgrade objects which have Briefing Tracking Profile Applet - Product marked as 'Upgrade Anc'`

Review errors through the Upgrade Applications Objects List screen. To get to the Upgrade Applications Objects List screen, from the Screens menu in Siebel Tools, choose Application Upgrader > Upgrade Application Objects List.

To view all errors encountered during the merge, query for `ERROR::` in the Status field of the Object Differences applet (second applet on this view). Typically, all error records will be listed at the top.

Determining the Cause of Merge Errors

If errors are located, you need to review the merge.txt file. The merge.txt file is located in the `Tools\bin` directory. This file contains information that is logged while the merge is in progress. As errors are encountered during the merge, they are logged in the merge.txt file, preceded with an exclamation point (`!`).

NOTE: Each time you run the merge process, the merge.txt file is incremented and renamed with numbers appended to it. For this reason, you may find that your file has a name such as merge1.txt.

To search for logged errors in this file, search for exclamation points (`!`) throughout the file.

CAUTION: The merge.txt file can be used to figure out the cause of the error. In case of performance issues, this file can help you to determine where the time is being taken. Do not proceed if you encounter any errors as a result of the merge. See "Recovering from a Failed Merge" on page 187. If you have any questions or require assistance in resolving these errors, please contact Technical Support or Professional Services.

Reviewing Merge Process Outcome

After performing the merge, review the merge process outcome.

To review the overall outcome of the merge process

1 Navigate to the Application Upgrades object list view by choosing Tools > Upgrade > Upgrade Application. (You can also access the Application Upgrades object list view from the Screens menu by choosing Screens > Application Upgrade.)

   The Application Upgrades applet at the top of this view lists all of the merge processes performed by the current user.

   The Status field for a successful merge should read "Complete."

2 If an error occurs, your merge has failed and you need to perform procedures to recover from a failed merge. To recover from a failed merge, see "Recovering from a Failed Merge" on page 187.

Reviewing Object Differences

After reviewing the merge process outcome, review object differences.

To review object differences

The Applications Upgrade object list view should still be displayed from the previous step. If the Applications Upgrade object list view is not displayed, navigate to it by choosing Tools > Upgrade > Upgrade Application.

1. From the Applications Upgrade object list view, click the Object Differences applet.

   The Object Differences list displays new or deleted objects, or objects that had altered attributes. By default, the most important object differences are shown first, with errors appearing at the top of the list.

2. For each record, review the status field for errors. (You must tab to the right of the applet to see the status field.)

3. Query the applet for objects that do not have a check mark in the Attribute column; these are the new or deleted objects.

   The various In… columns show which repositories the object existed in before the repository merge. The Add to New Customer Repository column indicates whether the object was merged into the New Customer Repository.

There are two types of object differences that must be resolved:

- Objects you previously deleted
- Objects deleted by Siebel eBusiness Applications

Customer-deleted objects. Any object that has a check mark in the “In Prior x.x Siebel Repository” and “Added to New Customer Repository” columns, but not in “In Prior Customer Repository,” was deleted during your customization of the Prior Customer Repository and automatically added back to the New Customer Repository. Typically, adding these objects back into your new repository does not cause a problem with your upgraded configuration. However, you need to examine the use of these objects carefully to make sure that this is the case.

Siebel-deleted objects. Any object that has a check mark in the “In Prior x.x Siebel Repository” and “In Prior Customer Repository” columns, but not in the “In New Siebel Repository” or “Added to New Customer Repository” columns, is obsolete in Release 7.5. Typically, when a new version of a similar object is introduced as part of a new release of Siebel eBusiness Applications, you must modify your upgraded configuration to make use of the new object.

The fields in the Object Differences list indicate common types of object definition differences. Table 15 lists four common differences.

Table 15. Common Object Definition Differences from the Object Differences List

Common Object Type Difference	Conflict	Add to New Customized	In Prior Standard	In Prior Customized	In New Standard
Indicates an object definition that became obsolete in Release 7.5.			✓	✓	
Indicates an object definition added by the customer.		✓		✓	
Indicates an object definition that has been modified. The Attribute Differences list will then show the property differences.	✓	✓	✓	✓	✓
Indicates a new Release 7.5 object definition. This difference will not normally be shown. To show these object definitions, the Skip Logging... check box on the Advanced Merge Options dialog box must be turned off.					✓

Conflict Resolution for the Repository Merge

CAUTION: Repository merge procedures only apply to development environment upgrades. If you are performing a production environment upgrade, skip to “Upgrading the Custom Database Schema” on page 189.

Reviewing Object Attribute Differences

Object attribute differences are shown in the Attribute Differences applets of the Application Upgrades Object List and Application Upgrades Attributes List views. The Object List view shows the differences for the currently selected object; the Attributes List view shows differences for all objects. You need to review any attribute differences that have a check mark in the Conflict column.

In either view, query all entries that have a check mark in the Conflict column, and ignore attributes of the following types:

- Left
- Right
- Top
- Height
- Width

These are screen layout attributes that can be detected when running the Siebel client and corrected after you have completed the upgrade process.

The Resolution column shows whether the Standard Value (shown in the “In Prior x.x Siebel Repository” column) or the Custom Value (shown in the “Prior Customer Repository” column) was used as the object attribute in the New Customer Repository.

You can change the resolution for any conflict by entering a check mark for the Override flag. This automatically sets the attribute in the New Customer Repository to the opposite value.

CAUTION: Do not rerun the repository merge into the New Customer Repository after entering the Override flag check mark. If you run the merge again, you will remove the Override attribute.

You can override attribute differences only as long as each repository used during the merge process remains in its current state in the database server. Resolve all attribute conflicts before continuing with the upgrade process.

Examples of Conflict Resolution

The following examples demonstrate how conflicts are resolved during the merge.

A customer change that would be preserved by the merge. If the value in Prior Customer Repository is unique, and the values in Prior Siebel Repository and New Siebel Repository are the same, the customer change will be preserved in the New Customer Repository.

For example, if the following statements are true,

Prior Siebel Repository has value `50` for the property `height` of applet `X`
Prior Customer Repository has value `25` for the property `height` of applet `X`
New Siebel Repository has value `50` for the property `height` of applet `X`

then the value of `25` will be set in the New Customer Repository.

A customer change that would be overridden by the merge. If the values in all three repositories are different, then the default value from the previous release was changed by the customer, but also changed in the new release. The two updated values result in a conflict. In 90% of cases where there is a conflict, the value from New Siebel Repository will override the value in the Prior Customer Repository.

For example, if the following statements are true,

Prior Siebel Repository has value `20` for the property `height` of applet `X`
Prior Customer Repository has value `25` for the property `height` of applet `X`
New Siebel Repository has value `50` for the property `height` of applet `X`

then the value from the New Siebel Repository will be merged into the New Customer Repository.

Certain attributes determine the 10% of cases where the value from the Prior Customer Repository will override the value in the New Siebel Repository. View the attribute under object type `Type` in Tools.

- If the `StandardWins` flag is set to `Y`, the New Siebel Repository value is chosen.

CAUTION: If the `StandardWins` flag is set to `N`, then the Prior Customer Repository value is chosen. The `StandardWins` settings delivered “out-of-the-box” are highly recommended. Avoid changing the `StandardWins` settings.

Troubleshooting

CAUTION: Repository merge procedures only apply to development environment upgrades. If you are performing a production environment upgrade, skip to "Upgrading the Custom Database Schema" on page 189.

Recovering from a Failed Merge

If the repository merge process fails for any reason (for example, due to a lack of virtual memory on the Siebel Tools workstation, or a database error), perform the following steps to restore your pre-merge database:

- Restore your database from the backup that you took after you upgraded the Siebel database schema (upgrep). See "Making a Backup of the Database Repository" on page 158.
- Fix the errors that caused the merge to fail.
- Prepare the prior customer repository for the merge if your database backup does not include migrated strings, merged labels and fields, and merged Web templates. See "Preparing the Prior Customer Repository for the Merge" on page 160.
- Rerun the merge. See "Performing a Repository Merge" on page 175.

Generating EIM Temporary Columns

CAUTION: This procedure only applies to development environment upgrades. If you are performing a production environment upgrade, skip to "Upgrading the Custom Database Schema" on page 189.

The repository merge process does not preserve EIM processing columns for custom mappings. After the completion of a successful merge, you need to execute the steps below to generate the missing custom EIM processing columns.

To generate EIM temporary columns

1. Open Tools on "New Customer Repository."
2. Go to the Tools menu > Upgrade > Generate EIM Processing Columns.

3 At the dialogue box, click OK to generate EIM processing columns for custom mappings.

4 Verify that this process has completed successfully. To do this, in the Object Explorer window in Siebel Tools, select the Repository object that corresponds to the New Customer Repository and check that the Comment field shows `UpgEimCol`.

NOTE: The repository that you run this process on must be the one that you use to upgrade the Custom Database Schema (upgphys).

Upgrading the Custom Database Schema

The configuration utility that you ran in "Upgrading the Siebel Database Schema" on page 147 updated the Siebel Database Schema. You will now use the Database Server Configuration Wizard to apply these changes to the custom database schema to upgrade it to the new version.

If you are upgrading a development environment, the repository on which you run the Custom Database Schema upgrade must be the same repository that you prepared for the merge in "Generating EIM Temporary Columns" on page 187.

Before upgrading the custom database schema, you should take a backup of your Siebel database.

NOTE: The custom database schema upgrade may complete in less than one hour.

To upgrade the custom database schema (upgphys) on supported UNIX platforms

1. Verify that all servers are stopped:

   - Stop the Siebel Gateway by typing: `stop ns`
   - Stop the Siebel Servers by typing: `stop server ALL`

2. Depending on your shell, source environment variables from *`$SIEBEL_ROOT`* by typing:

   Korn shell

   ```
   . siebenv.sh
   ```

   C shell

   ```
   source siebenv.csh
   ```

3. Set the following environment variables:

   - `SIEBEL_ROOT` should be the path of your Siebel eBusiness Application installation directory.

- `LANGUAGE` should be set to the language in which the Configuration Wizard prompts appear; for example, `enu` for U.S. English.

If either of these values is incorrect or empty, reset them using one of the following commands for the shell you are using:

C shell

- ❑ `setenv LANGUAGE` *`enu (`*where *`enu represents your display language)`*
- ❑ `setenv SIEBEL_ENTERPRISE` *`Enterprise Name`*

Korn shell

- ❑ `LANGUAGE=enu ; export LANGUAGE`
- ❑ `SIEBEL_ENTERPRISE=`*`Enterprise Name`*` ; export SIEBEL_ENTERPRISE`

4 Navigate to `$`*`SIEBEL_ROOT`*`/bin` and enter:

```
dbsrvr_config.ksh
```

This launches the Database Server Configuration Wizard.

5 Review the values of the following environment variables and confirm whether or not the settings are correct by entering either `Y` or `N`.

- ❑ `SIEBEL_ROOT`
- ❑ `LANGUAGE`

NOTE: If either the `SIEBEL_ROOT` or `LANGUAGE` value is not set or is incorrect, you must correct them before proceeding.

6 In the Siebel Server Directory screen, specify the path of your Siebel Server root directory, or accept the default by pressing Enter.

7 In the Siebel Database Server Directory screen, specify the path of your database server root directory, or accept the default by pressing Enter.

The RDBMS Platform screen appears.

8 In the RDBMS Platform screen, enter the number that corresponds to your database platform.

1 - IBM DB2 UDB for Unix and Windows.

2 - Oracle.

The Siebel Database Operation menu appears.

9 From the Siebel Database Operation menu, choose Upgrade Database (`2`).

10 From the Upgrade Options menu, choose Upgrade Custom Database Schema (upgphys) `(2).`

The Environment Type screen appears.

11 From the Environment Type screen, choose the number that corresponds to the environment that you are upgrading:

```
1 - Development.

2 - Production.
```

Which screen appears next depends on the application that you are upgrading:

- **Siebel eBusiness Applications.** The Current Siebel Version screen appears. Skip Step 12 and continue to Step 13.
- **Siebel Industry Applications or Siebel Financial Services Applications.** The Siebel Industry Vertical screen appears. Continue with Step 12.

12 If you are upgrading a Siebel Industry Application or Siebel Financial Services Application, choose the application from which you are upgrading. Carefully read the descriptions of the available applications. The choices available to you are constrained by the database platform you selected. The following applications may appear:

NOTE: Before choosing your application, see *Siebel System Requirements and Supported Platforms* to verify the supported upgrade paths for your deployment.

- **Siebel Automotive.** Choose this if you are upgrading from Release 6.x of Siebel Automotive.

- **Siebel Communication Media Energy.** Choose this if you are upgrading from Release 5.x or 6.x of Siebel Communications, Media, and Energy.
- **Siebel Consumer Sector.** Choose this if you upgrading from Release 5.x or 6.x of Siebel Consumer Sector.
- **Siebel Life Sciences.** Choose this if you are upgrading from Release 5.x or 6.x of Siebel Life Sciences.
- **Siebel Industry Solutions.** Choose this if you are upgrading from Release 7.0.3 or 7.0.4 of any of the following applications: Siebel Consumer Sector, Siebel Life Sciences, Siebel Communications, Media, and Energy, Siebel Automotive.
- **Siebel Financial Services.** Choose this if you are upgrading from Release 5.x, 6.x, or 7.x of Siebel Financial Services.
- **Siebel Application.** Choose this if you are upgrading from Release 7.5 of Siebel eBusiness Applications to Release 7.5 of a Siebel Industry Application.

13 In the Current Siebel Version screen, select the version of Siebel you are upgrading from.

The Language Selection screen appears.

14 Enter the number that corresponds to the appropriate language.

15 Enter the ODBC datasource name; for example, `$SIEBEL_ROOT_siebel`.

NOTE: The data source is created automatically by the Siebel Server installation, using the format $*`SIEBEL_ROOT_EnterpriseName`*. To find the name of your ODBC data source, type: `vi $ODBCINI`

---

16 Enter the Database User Name and Password for your database:

a **Database User Name.** User name of the Siebel administrator, for example, `sadmin`.

b **Database Password.** Password for the Siebel administrator, for example, `db2`.

17 Enter the database tableowner name (this is the account that will own the Siebel objects):

Database Tableowner. User name of the tableowner, for example, `siebel`.

Database Tableowner Password. Password for the tableowner, for example, `db2`.

18 Enter the number that corresponds to the platform of your database server operating system:

```
1- Windows

2- Solaris

3- AIX

4- HP-UX
```

19 The prompts that appear next may vary, depending on which database platform you use. See the following table for the prompts that you may receive next. Enter the values that you recorded on your Upgrade Planning Worksheet.

NOTE: Use underscores rather than spaces; these values are case-sensitive.

Field	Value
Index Space	■ DB2 UDB: The name you give to your 4-KB index space for tables ■ Oracle: The name you gave to your index area
4KB Table Space (DB2 UDB only)	■ DB2 UDB: The name you give to your 4-KB table space Note that you must use underscores rather than spaces.
8KB Table Space (DB2 UDB upgrades from 5.x only)	■ DB2 UDB: The name you gave to your 8-KB table space
16KB Table Space (DB2 UDB only)	■ DB2 UDB: 16 KB Table space name
32KB Table Space (DB2 UDB only)	■ DB2 UDB: 32 KB Table space name
Table Space Name	■ Oracle: The name you gave to your data area

After you enter the requested values, the utility displays the values entered so far.

20 Review the values for accuracy against the values that you recorded in your copy of Appendix A, “Upgrade Planning Worksheet.”

- If the values are correct, launch the Siebel Upgrade Wizard (`srvrupgwiz`) by entering `Y`. The Siebel Upgrade Wizard is explained in "Reviewing the Upgrade Log Files" on page 153.

  NOTE: If a program or system error occurs and you need to rerun the Siebel Upgrade Wizard, you can do so, starting at the point at which the wizard failed, by navigating to $*SIEBEL_ROOT*/bin and typing:

  ```
  srvrupgwiz /m master_upgrade_option_upgrade_type_
  version.ucf
  ```

  For example, to restart a development environment upgrade from Release 6.0.1 to Release 7.x, type:

  ```
  srvrupgwiz /m master_upgphys_dev_601.ucf
  ```

- If you need to change the values you previously entered, rerun the Configuration Wizard by entering `dbsrvr_config.ksh`. Rerunning the Configuration Wizard will allow you to reconfigure with different values.

The Siebel Upgrade Wizard displays a message when the upgrade is complete.

CAUTION: The upgrade process executes in a UNIX shell. Do not click inside this shell while the upgrade is running. If you accidentally click inside the UNIX shell, the process will stop executing and you will need to restart the upgrade.

Reviewing the Upgrade Log Files

The repository upgrade wizard creates several log files, such as `srvrupgwiz1.log`, `srvrupgwiz2.log` (the name of the log file will increment for subsequent log files) within the *`SIEBEL_ROOT`*`/log` directory. You must review the log files carefully for unacceptable errors. Subsequent log files are created when the upgrade wizard encounters a problem and the user attempts a retry.

To review the log files, see "Reviewing the Upgrade Log Files" on page 153.

CAUTION: Do not proceed with the upgrade until unacceptable errors have been corrected. If you cannot correct the error, contact Siebel Technical Support or Professional Services to report the error in detail.

Troubleshooting the Custom Database Schema Upgrade

Typical problems that might occur during the upgrade of your custom database schema (upgphys) could result from a lack of storage space or insufficient user privileges.

Recovering from a Failed Custom Database Schema Upgrade

If the repository upgrade fails due to insufficient space allocated on the database, you must complete the following procedures.

To recover from a failed custom database schema upgrade

1. Back up your complete set of log files, from the beginning of the upgrade process to the point at which it stopped, to another directory.

2. Read the `srvrupgwiz1.log` file and associated log files to determine the failure. See "Reviewing the Upgrade Log Files" on page 196.

3. Take the necessary corrective action. Depending on the errors that you find, you may need to ask your DBA to extend the database.

4 Resume the upgrade wizard. It will continue from the point at which it failed.

NOTE: To resume the upgrade wizard, navigate to $*SIEBEL_ROOT*/bin and type:

```
srvrupgwiz /m master_upgrade_option_upgrade_type_
version.ucf
```

For example, to restart a development environment upgrade from Release 6.0.1 to Release 7.x, type:

```
srvrupgwiz /m master_upgphys_dev_601.ucf
```

Restarting the Upgrade

The Siebel Upgrade Wizard is restartable at most stages within the upgrade process. If the Siebel Upgrade Wizard encounters an error during the upgrade process, it will stop at that point. If your upgrade stops due to an error, you must carefully review the log files to make sure that your upgrade has completed successfully up to that point. Once you have verified this and resolved the failure, you may restart the upgrade. The upgrade will continue from the last step that completed successfully.

CAUTION: Before you restart the upgrade (after any break in the upgrade process) back up your complete set of log files, from the beginning of the process to the point at which it stopped, to another directory. This will maintain a complete record of your log files, and prevent your previous log files from being overwritten, which might prevent accurate diagnosis of the reason for the break in the upgrade.

Launching the Siebel Upgrade Wizard

The Siebel Upgrade Wizard launches from the final prompt of the Server Database Configuration Utility. However, if a program or system error occurs and you need to rerun the Siebel Upgrade Wizard, you can do so, starting at the point at which the wizard failed, by using the following procedure.

To relaunch the Siebel Upgrade Wizard or to restart the upgrade

1 Navigate to $*SIEBEL_ROOT*/bin and type:

```
srvrupgwiz /m master_upgrade_option_upgrade_type_version.ucf
```

For example, to restart a production environment upgrade from Release 6.0.1 to Release 7.x, type the following command:

```
srvrupgwiz /m master_upgphys_prod_601.ucf.
```

2 To begin the upgrade of your repository, at the prompt, press Enter.

The Siebel Upgrade Wizard will notify you when the upgrade process is complete.

You have finished upgrading your repository and are ready to review the log files for errors. See "Reviewing the Upgrade Log Files" on page 196.

Adding New License Keys

CAUTION: Skip this procedure if you are upgrading a development environment. In development environment upgrades this procedure was performed after upgrade of the *Siebel database schema*. In production environment upgrades this procedure is performed after upgrade of the *custom database schema*.

With the new Release of Siebel eBusiness Applications, you received one or more license keys that must be added to the production database. You must add all of the new license keys to enable the new release of the Siebel eBusiness Applications, which you will use in the next steps of the upgrade process.

To add new license keys

1 Start Siebel Tools version 7.x from a production workstation and log on to the database server as the Siebel administrator.

2 Add your new license keys.

Continuing with the Upgrade

If you have successfully completed the upgrade tasks described in this chapter, go to Chapter 5, “Postupgrade Tasks.”

Postupgrade Tasks 5

This chapter describes the tasks that must be performed after you perform an upgrade of either your development environment or your production environment. To better understand which phase of the upgrade process you are in and what tasks still remain, refer to Figure 7. Figure 7 illustrates where you are in the upgrade process depending on whether you are performing a development or production environment upgrade.

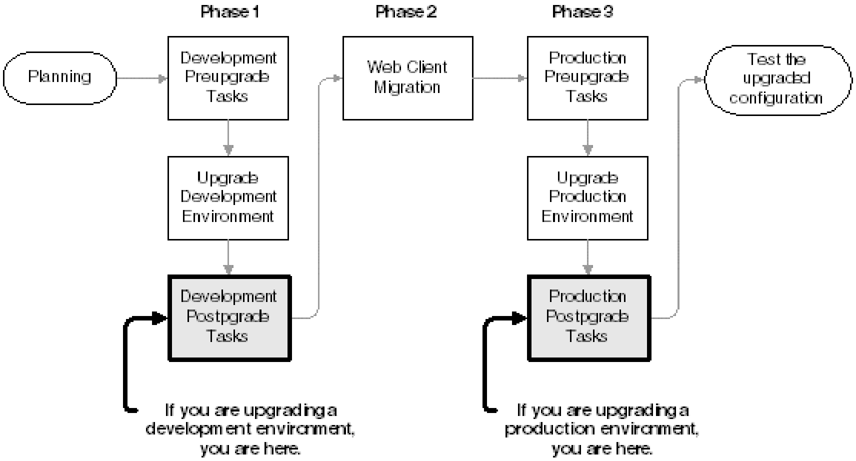

Phase 1
Phase 2
Phase 3
Planning
Development Preupgrade Tasks
Web Client Migration
Production Preupgrade Tasks
Test the upgraded configuration
Upgrade Development Environment
Upgrade Production Environment
Development Postpgrade Tasks
Production Postpgrade Tasks
If you are upgrading a development environment, you are here.
If you are upgrading a production environment, you are here.

Figure 7. You Are Here

NOTE: If a procedure is specific to one environment, this will be indicated by a caution at the beginning of the procedure; for example, CAUTION: This procedure only applies to development environment upgrades.

A checklist of postupgrade tasks is provided in Table 16. Perform the procedures for the environment you are upgrading.

- **Development environment upgrades**. Conduct postupgrade procedures for development environments, then proceed to Chapter 6, "Migrating the Client Configuration to the Web Client." Skip the procedures that only apply to production environments.
- **Production environment upgrades**. Conduct postupgrade procedures for production environments. Skip procedures that only apply to development environments. When you have completed postupgrade procedures, you have completed your upgrade. However, if you did not perform Web Client migration as part of your development environment upgrade, you must perform it after you complete postupgrade tasks for the production environment. Go to Chapter 6, "Migrating the Client Configuration to the Web Client."

Table 16. Postupgrade Tasks

		Environment	
Postupgrade Tasks		Dev	Prod
1	Drop DB2 8-KB tablespaces and buffers. See "Dropping DB2 8-KB Tablespaces and Buffers" on page 205.	✔	✔
2	Manually archive the log files. See "Manually Archiving the Log Files" on page 205.	✔	✔
3	Update file system directory. See "Updating the File System Directory" on page 205.	✔	✔
4	Update file system attachments. See "Updating File System Attachments" on page 206.	✔	✔
5	Migrate custom columns on obsolete tables. See "Migrating Custom Columns on Obsolete Tables" on page 209.	✔	
6	Run update statements. See "Running Update Statements" on page 212.	✔	✔
7	Reapply custom extensions and FK columns on obsolete tables. See "Reapplying Custom Extensions and FK Columns on Obsolete Tables" on page 213	✔	
8	Migrate custom business components and resolve business component and join conflicts. See "Migrating Custom Business Components and Resolving Conflicts" on page 217.	✔	
9	Identify Obsolete Objects. (Optional) See "Identifying Obsolete Objects" on page 219.	✔	

Table 16. Postupgrade Tasks

Postupgrade Tasks	Environment	
	Dev	Prod
10 Generate reporting relationships. See "Generating Reporting Relationships" on page 220.	✔	✔
11 Set visibility modes for access control. See "Setting Visibility Modes for Access Control" on page 222.	✔	
12 Resolve duplicate EIM mappings. See "Resolving Duplicate EIM Mappings" on page 226.	✔	
13 Safeguard the Custom Repository export file. See "Safeguarding the New Custom Repository Export File" on page 227.	✔	
14 Produce a new custom configuration file. See "Producing a New Custom Configuration File" on page 229.	✔	
15 Regenerate the database template file. See "Regenerating the Database Template File" on page 230.	✔	
16 Extract Developers or Siebel Tools Clients. See "Extracting Developers or Siebel Tools Clients" on page 230.	✔	
17 Test the Upgraded Configuration. See "Testing the Upgraded Configuration" on page 230.	✔	
18 Validate dock objects and rule definitions. See "Validating Dock Objects and Rule Definitions" on page 231.	✔	✔
19 Run statistics. See "Run Statistics" on page 233.	✔	✔
20 Reorganize the P1 index for DB2 UDB. See "Reorganize the P1 Index for DB2 UDB" on page 233.	✔	✔
21 Recreate server component definitions. See "Recreate Server Component Definitions" on page 233.	✔	✔
22 Reset database server configuration parameters. See "Reset Database Server Configuration Parameters" on page 233.	✔	✔
23 Clean up repository files. See "Clean Up Your Repository Files" on page 233.	✔	
24 Perform postupgrade task for Call Center. See "Call Center" on page 234.	✔	✔

Table 16. Postupgrade Tasks

Postupgrade Tasks	Environment	
	Dev	Prod
25 Perform postupgrade task for eBusiness Application Integration (EAI). See “Enterprise Application Integration (eAI)” on page 235.	✔	
26 Perform postupgrade task for Siebel Financial Services. See “Siebel Financial Services” on page 235.	✔	✔
27 Perform postupgrade task for Siebel Life Sciences. See “Siebel Life Sciences” on page 239.	✔	✔
28 Perform postupgrade task for Marketing. See “Marketing” on page 240.	✔	✔
29 Perform postupgrade task for Siebel Purchase Orders. See “Siebel Purchase Orders” on page 240.		✔
30 Perform postupgrade task for Quotes. See “Quotes” on page 241.	✔	✔
31 Perform postupgrade task for Workflow. See “Siebel Workflow” on page 242.	✔	✔
32 Perform postupgrade task for Siebel .eConfigurator. See “Siebel eConfigurator” on page 243	✔	✔
33 Upgrade your encryption method to RC2 after your upgrade is complete and before you deploy your application. See “Upgrading to RC2 Encryption” on page 244.	✔	✔
34 Set up your environment for support global time zone. See “Setting Up Your Environment to Support Global Time Zone” on page 251.	✔	✔
35 Review planning considerations for migration from a non-Unicode code page to Unicode. See “Migrating to Unicode” on page 253.	✔	✔
36 Upgrade regional servers. See “Upgrading Regional Servers” on page 256.		✔
37 Upgrade mobile and connected clients. See “Upgrading Mobile and Connected Clients” on page 256.		✔

Postupgrade Tasks for All RDBMS Development Environments

Perform the following postupgrade tasks for all RDBMS development environments.

Dropping DB2 8-KB Tablespaces and Buffers

If you are upgrading a database under Windows or UNIX, you should now drop your 8-KB tablespace, 8-KB temporary tablespace, and 8-KB bufferpool. Before dropping your 8-KB tablespace, check for the existence of any tables in it by running the following SQL statement:

```
select name from sysibm.systables where TBSPACE='TBS_8K'
```

CAUTION: Check your 8-KB tablespace to make sure that all objects have been migrated to your new 16-KB tablespace before dropping the 8-KB tablespace. Otherwise, you will lose this data.

Manually Archiving the Log Files

After a successful installation and upgrade, you must save and archive the log files located in the `siebsrvr_root/log` directory. This is a manual process.

By default, only nine (9) log files are retained for subsequent retries of the upgrade wizard. After nine log files have been created, when the upgrade wizard is rerun, it will overwrite log files beginning with the earliest one created and recycle the rest as necessary.

The number of log files retained can be increased by resetting the `siebel_log_archive` environment variable to 20, for example, to retain twenty (20) log files.

Updating the File System Directory

In your previous installation, the file system had a flat structure with one directory. The Release 7.x file system contains subdirectories. During the upgrade of your production environment, Siebel Anywhere will look for certain files in the file system subdirectories, but these files only exist in the root file system directory.

Therefore, you need to create the appropriate subdirectories in your previous installation directory structure and then copy the required files from the root file system directory to the new subdirectories.

To make file attachments accessible by Siebel eBusiness Applications

1 In the existing file system structure, create an `/att` subdirectory if it does not already exist; for example, /*siebfile*/att

where:

siebfile is your root file system directory.

2 Copy all files located under the /*siebfile* directory to the //*siebfile/att* directory so that all file attachments will be accessible by Siebel eBusiness Applications.

3 Verify that files have copied correctly to the //*siebfile/*att directory. After this has been verified, clean up the file system.

Updating File System Attachments

During the Siebel database upgrade, data from the inactive table `S_EMPLOYEE` was migrated to `S_CONTACT, S_USER, S_EMP_PER` and data from `S_ORG_INT` was migrated to `S_ORG_EXT`. Attachments in `S_ORG_INT_ATT` were migrated to `S_ACCNT_ATT` and attachments in `S_EMPLOYEE_ATT` were migrated to `S_CONTACT_ATT`.

NOTE: If your upgrade to Release 7.5 was from Release 7.0.3 or 7.0.4, files from `S_WEB_CNTNT` were migrated to `S_CB_ASSET_VER`.

Data migration affects the physical file structure of the Siebel File System. Therefore, you need to run a utility to rename file attachments which correspond to inactive tables for Release 7.x so that they will be accessible by Siebel eBusiness Applications. For example, this utility will copy and rename all files named `S_EMPLOYEE*.SAF` to `S_CONTACT*.SAF` and all files named `S_ORG_INT_ATT*.SAF` to `S_ACCNT*.SAF` so that they correspond to new table names.

To upgrade file attachments

- Run `chng_file_sys.ksh`. From *`SIEBSRVR_ROOT/bin,`* enter the following command:

  ```
  chng_file_sys.ksh -s Source_Table -t Target_Table
  -f "File_System"
  ```

  where:

  - *`Source_Table = name of the original, inactive, table`*
  - *`Target_Table = name of the new table to which the original data was migrated`*
  - *`"File_System" = name of the directory where the file system resides (entered inside quotation marks)`*

  `for example,`

  ```
  chng_file_sys.ksh -s S_EMPLOYEE -t S_CONTACT
  -f "D:/DBSRVR_ROOT/Files"
  ```

Review the renamed files carefully to verify that they can be accessed by Siebel eBusiness Applications.

CAUTION: Make sure that attachment files for obsolete tables are renamed or copied to alternate locations.

For example, since `S_EMPLOYEE_ATT` is migrated to `S_CONTACT_ATT`, you need to rename a file such as `S_EMPLOYEE_12-1ABC.SAF` to `S_CONTACT_12-1ABC.SAF`.

If you upgraded from Release 7.0.3 or 7.0.4, data in S_LIT was migrated to S_CB_ASSET_VER; therefore, you need to migrate files associated with migrated records. Run the following utility to copy the necessary files named S_LIT*.SAF to S_CB_ASSET_VER*.SAF so that the files correspond to the new table name.

To upgrade S_LIT files to S_CB_ASSET_VER files

- Run `file_upg_mm.ksh`. From *`SIEBSRVR_ROOT`*`/bin`, enter the following command:

```
file_upg_mm.ksh Odbc_Source User_Name Password Table_Owner
File_System_Location Siebel_Home_Directory
Siebel_Db_Server_Home_Directory
```

where:

- *Odbc_Source* = the ODBC source of the database
- *User_Name* = the database user name
- *Password* = the password for the database user name
- *Table_Owner* = the database table owner (if your implementation is on iSeries, you can use the value provided for *User_Name*)
- *"File_System_Location"* = the directory where the file system resides (entered inside quotation marks)
- *"Siebel_Home_Directory"* = the directory where Siebel Server is installed (entered inside quotation marks)
- *"Siebel_Db_Server_Home_Directory"* = the directory where Siebel Database Server is installed (entered inside quotation marks)

for example,

```
file_upg_mm.ksh SEBL sadmin sadminpw SIEBEL "D:/DBSRVR_ROOT/
Files" "D:/SEBLSRVR_ROOT" "D:/DBSRVR_ROOT"
```

Review the renamed files carefully to verify that they can be accessed by Siebel eBusiness Applications.

Migrating Custom Columns on Obsolete Tables

CAUTION: This procedure only applies to development environment upgrades. However, the output of this procedure will also be used in the production environment.

Customer extension columns on obsolete tables need to be identified and manually migrated after the upgrade. However, Siebel columns on obsolete tables are reconfigured and migrated during the upgrade and therefore do not need to be manually migrated. Custom extension columns on tables that are upgraded (not obsolete) are also retained during the upgrade.

Critical obsolete tables are listed in Table 17. Any customer extension columns based on these tables must be manually migrated.

Table 17. Repository Tables That Are Obsolete in Release 7.x

Previous Table	Suggested New Table
S_EMPLOYEE	S_CONTACT, S_USER, S_EMP_PER
S_EMP_POSTN	S_PARTY_PER
S_ORG_INT	S_ORG_EXT, S_BU
S_POSTN_RPT_REL	S_PARTY_RPT_REL

A script, `rpt_obs_tbl_custom_cols.sql`, is available to identify customer columns on obsolete tables and to generate SQL template update statements to migrate customer columns on obsolete tables to new tables during the development and production upgrade.

This script has two parts:

- Part 1

  Part 1 compares the Prior Standard Siebel Repository with the Prior Custom Repository and generates a report of custom columns on obsolete tables for Release 7.x that need to be manually migrated. For example, if you created custom columns on S_EMPLOYEE, S_ORG_INT, or S_EMP_POSTN, the report would contain a list of custom columns on these obsolete tables. Use the output of this section of the script to analyze and identify custom columns on obsolete tables that require manual migration. Run this portion of the script during the upgrade of your development environment. You will use the information generated by this part of the script for both your development and production environment upgrades.

- Part 2

  Part 2 generates SQL template update statements for migrating data from obsolete tables to new target tables. Before running these statements, carefully check that the target tables specified are accurate for your configuration. Modify the SQL template update statements as needed. After the upgrade of your development environment and production environment, run the update statements to migrate data to new tables.

To run rpt_obs_tbl_custom_cols.sql

1 At the odbcsql command, under the `$SIEBEL_ROOT/siebsrvr/bin` directory, invoke `rpt_obs_tbl_custom_cols.sql` by entering:

```
odbcsql /s "$ODBCDataSource" /u $TableOwner /p $TablePassword /
separator / /a /v /c rem /l $SiebelRoot/log/
rpt_obs_tbl_custom_cols.log $DbsrvrRoot/$DatabasePlatform/
rpt_obs_tbl_custom_cols.sql "$AncestorRepName"
```

Table 18. Command line flags for rpt_obs_tbl_custom_cols.sql

Flag	Parameter	Description	Required?
/s	*$ODBCDataSource*	ODBC source name	Y
/u	*$TableOwner*	User name to log into database	Y
/p	*$TablePassword*	Password to log into database	Y
/a	Not applicable	Turn on ODBC auto-commit for session	N
/v	Not applicable	Turn on statement printing at execute	N
/c	*rem*	Remark. Specify string which begins at comment (at beginning of line)	Y
/l	*$SIEBELRoot/log/rpt_obs_tbl_custom_cols.log*	Write errors and status to log file specified here	Y
Not applicable	*$AncestorRepName*	Name of prior standard Siebel repository	Y

Part one of the script compares the Prior Standard Siebel Repository with the Prior Customer Repository and generates a report that includes obsolete tables and custom columns mapped to obsolete tables. The second part of the script generates SQL template update statements that you can use to manually migrate your custom columns to new target tables.

2 Review the report generated by part 1 of the script and identify custom columns on obsolete tables. After the upgrade, you need to remap all custom columns listed in this report to new target tables.

3 Review the SQL template update statements generated by part 2 of the script.

Modify the SQL template update statements for your configuration. After you have modified them, you can run them after the upgrade to migrate custom columns For more information, see "Running Update Statements."

NOTE: Run these update statements after you upgrade your development environment and after you upgrade your production environment.

Running Update Statements

In the development environment, you ran the script `rpt_obs_tbl_custom_cols.sql` to identify custom columns on obsolete tables and to generate SQL template update statements. You can use the SQL template update statements generated by this script to migrate data in custom columns on obsolete tables to new tables. To do this, review and edit the SQL template update statements for your configuration. For example, you should check that all target tables specified in the update statements are accurate for your configuration. After you have edited the statements, run them.

To run the update statements

1 Navigate to `$DBSRVR_ROOT/log/rpt_obs_tbl_custom_cols.log` to access the SQL template update statements generated by `rpt_obs_tbl_custom_cols.sql`.

2 Edit the SQL template update statements as required for your configuration. You can use a simple text editor application to edit the statements.

3 Using ODBCSQL, or any client or vendor provided interface which can execute SQL, run the update statements against the upgraded database.

Reapplying Custom Extensions and FK Columns on Obsolete Tables

CAUTION: This procedure only applies to development environment upgrades. For production environment upgrades, skip this procedure and proceed to "Generating Reporting Relationships."

NOTE: Skip this procedure if you are performing an upgrade from Release 7.0.x to Release 7.5.

Several tables used in prior versions of Siebel eBusiness Applications are no longer used, or have been replaced by new tables in Release 7.x. The new configuration uses the new tables. You need to reapply the extensions and FK columns that you added to previous versions of the tables to the new tables.

The upgrade process generates a report that you must review for information about tables that are either obsolete or no longer in use for Release 7.x. This report, `xtndobstbl.txt`, lists the following:

- Custom columns in obsolete tables
- Custom FK columns pointing to obsolete tables
- EIM mappings for custom FK columns to access-control related obsolete tables
- Workflow columns by custom FK to obsolete tables
- Customer denormalized columns on obsolete tables
- Obsolete tables in Release 7.x. Each obsolete table is listed with one of three codes:
  - **Not Used.** These tables are not used in Release 7.x, but you may continue to use them; for example, S_ORG_PROD, S_PERIOD BU. These tables are supported as is (for instance, with docking or EIM).
  - **EOL (end of life).** These tables are not used in Release 7.x, and they will not be supported in future releases; for example, old S_*_IF tables.

- **Inactive.** These tables have been discontinued, and are not supported in Release 7.x. Extension columns and FK columns that reside on inactive tables must be moved to alternate tables. For example, custom extension columns on S_EMPLOYEE should be moved to S_CONTACT, S_USER, or S_EMP_PER. Likewise, if you are upgrading from a pre-7.x version of Siebel Industry Applications, custom extension columns on S_PER_ORG_UNIT need to be moved manually to S_PARTY_PER, and custom extension columns on S_CONTACT REL need to be moved manually to S_PARTY_REL.

You need to review this report for extension columns and FK columns that reside on obsolete tables, and therefore need to be moved to alternate tables. If no tables are listed in `xtndobstbl.txt`, no action is required. If this file lists any tables, their custom extensions and FK columns must be reapplied to Siebel eBusiness Applications Release 7.x tables using Siebel Tools. See *Siebel Tools Reference*.

Table 19 lists examples of previously used tables which are inactive in Release 7.x (you can no longer use these tables) and lists the suggested new tables to which custom extensions need to be reapplied. The new tables are recommendations only; the tables that you need to apply the extensions to might vary depending on their type and use. Contact Siebel Technical Services to validate the reapplication of extension columns and to review the steps necessary to migrate any extension column data to the new tables.

Table 19. Examples of Tables That Are Obsolete in Release 7.x

Previous Table	Suggested New Table
S_EMPLOYEE	S_CONTACT, S_USER, S_EMP_PER
S_EMP_POSTN	S_PARTY_PER
S_ORG_INT	S_ORG_EXT, S_BU
S_POSTN_RPT_REL	S_PARTY_RPT_REL

If you have created many custom extension columns on the tables `S_EMPLOYEE` or `S_ORG_INT`, both of which are no longer used in Release 7.x, the joins between the tables will not be accurate. This may result in SQL errors when you launch the Siebel client.

In such cases, using Siebel Tools, you need to manually create corresponding extension columns in the new target tables, and manually move the data to the new extension column on the new table before you continue migration of the application. You should then review the business component configuration to make sure that the client will operate properly.

You may need to do this in one of the following instances:

- Fields based on custom extension columns in `S_EMPLOYEE` or `S_ORG_EXT`
- Fields based on custom extension tables from `S_EMPLOYEE` or `S_ORG_INT` with or without join
- Custom joins to custom extension tables from `S_EMPLOYEE` or `S_ORG_INT`

If you review the `xtndobstbl.txt` file after you run the upgrade, you will find a list of fields that require your attention.

Table 20 lists examples of previously used tables that are no longer used in Release 7.x, but that you may want to continue to use.

Table 20. Examples of Tables That Are Not Used in Release 7.x

Previous Table	Suggested New Table
S_CRSE	S_SRC, S_SRC_EVT
S_CRSE_OFFR	S_SRC, S_SRC_EVT
S_CRSE_REG	S_SRC_REG
S_CTLG_CAT_REL	S_CTLG_CAT
S_DNB_CONTACT	S_PRSP_CONTACT
S_OPTY_PROD	S_REVN
S_TMSHT_LINE	S_TMSHT_ITEM, S_TMSHT_LN

Table 21 lists examples of tables which were unused in previous releases of Siebel eBusiness Applications, but are now used in Release 7.x.

Table 21. Examples of Previously Unused Tables That Are Used in Release 7.x

Now Used Table	Used to be...
S_ACT_EMP	S_EVT_ACT
S_ACT_CON	S_EVT_ACT

Migrating Custom Business Components and Resolving Conflicts

This procedure only applies to development environment upgrades from Release 5.x or Release 6.x to Release 7.5.

Skip this procedure if you are upgrading a production environment or if you are upgrading from Release 7.0.x to Release 7.5. After the Upgrade Siebel Database Schema phase, thoroughly review the post upgrade configuration to make sure that the object level definitions are preserved as expected.

The access control buscomp migration utility, upgcust.exe, is run on the Prior Customer Repository by the Upgrade Wizard during upgrade of the Siebel database schema. This utility reconfigures custom business component configurations on obsolete tables. However, you may need to perform manual steps to business component fields and joins, depending upon the complexity of your business component configuration, because in some cases, not all custom business component configurations are reconfigured by the access control buscomp migration utility. For example, complex customized business component configurations and customer business components which are based on customer extension columns or on Siebel columns in obsolete tables need to be identified and manually fixed after the upgrade.

During the upgrade, the access control buscomp migration utility, upgcust.exe, generates a log file titled `upgcust.log` that contains a list of business component joins and fields that need to be manually rectified. This particular log file, along with others generated by the upgrade process, can be found under *`SIEBEL_ROOT`*`/log`.

The log file contains two distinct sections:

- **Part 1**

  For extension columns on obsolete tables such as S_EMPLOYEE and S_ORG_INT, you need to reimplement the extension columns on the replacement tables. After you have done this, review the business component definition to verify proper operation.

  The report generated by part 1 of the log file provides a list of the business component fields that are based on custom extension columns in obsolete tables such as S_EMPLOYEE and S_ORG_INT. This list displays the following properties:

  - Business Component Name
  - Field Name

- Column Name

  The table S_EMPLOYEE is migrated to three tables, S_CONTACT, S_EMP_PER and S_USER. After you have determined and implemented the approach for previously defined custom extension columns on obsolete tables, you should manually configure the business component field to reference that database column. Any manual reconfiguration must be done in the New Customer Repository after the upgrade has been run.

- **Part 2**

  After the repository merge has been run, there may be inconsistencies in the join names and joins set at the field level due to merge behavior. If you do not resolve these discrepancies, it is likely that the application configuration will result in errors or will result in incorrect behavior. The report generated by part 2 of the log file provides a list of joins that were not updated during the merge process. This list displays the following properties:

  - Business Component Name
  - Join Name

  With this list, you will need to go to each of the Business Component definitions and manually change the join name from the current value to the value listed in the report. Make sure that there is consistency between the joins as defined by name and the joins defined for each of the fields. Again, any manual configuration must be done in the New Customer Repository after the upgrade has been run. For more information about working with Business Component definitions and joins, see *Siebel Tools Online Help*.

Identifying Obsolete Objects

CAUTION: This procedure only applies to development environment upgrades.

In Release 7.5, many objects have become obsolete. After performing the repository merge, you can generate a list of obsolete objects using Siebel Tools. Objects that were available in the Prior Standard repository are compared with the objects that are available in the New Standard repository. All objects that were available in the Prior Standard repository but are not available in the New Standard repository are obsolete in Release 7.5.

NOTE: This procedure is optional.

To generate a list of obsolete objects

1. On the Siebel Tools workstation on which you performed your successful repository merge, select Screens > Application Upgrader > Application Upgrade Object List.

2. In the Application Upgrade list, select the record of the successful merge.

3. In the Object Differences list, click Query.

4. Enter your query criteria in the Object Differences list:

   - Click the In Prior Standard check box so that a checkmark appears in the check box.

   - Click the In New Standard check box and then click the check box again so that a check does not appear in the check box.

   - Click the In Prior Customized check box so that a checkmark appears in the check box.

   - Click the Attribute check box and then click the check box again so that a check does not appear in the check box.

5. Press Enter to run the query.

   All obsolete objects in Release 7.5 appear in the Object Differences list. You can filter the objects displayed by using the Top Parent Type and Object Type fields.

Generating Reporting Relationships

In Release 7.x, there are three visibility hierarchies—position, organization, and access groups. These hierarchies are denormalized and maintained in the table S_PARTY_RPT_REL. These denormalized hierarchies are necessary for executing visibility modes that go up or down a hierarchy. For example:

- **Manager view mode.** “My Team's Accounts View” displays all accounts on which a manager and their subordinates are working.
- **Suborganizations view mode.** “All Contacts across My Organizations View” displays all contacts that are associated to either my organization or any of my organization’s sub-organizations.

The generate reporting relationships process rebuilds the denormalized relationships in the S_PARTY_RPT_REL table so that the hierarchical view modes will display the correct information. The basic operation of the function is to empty the S_PARTY_RPT_REL table and then walk through each S_PARTY record to recreate the denormalized hierarchical structures in the table. This process generates a large number of transactions for Siebel Remote users and regional nodes.

NOTE: This operation is time and cpu/memory intensive. The process may take several minutes, depending on the size and complexity of your organizational structures. You should not perform this when you are running other memory-intensive processes.

The Generate Reporting Relationships process needs to be executed after the upgrade to Release 7.x and whenever the denormalized hierarchy structure (S_PARTY_RPT_REL) becomes out of sync with the data in the normalized tables (S_PARTY). The following situations may cause these tables to become out of sync:

- After upgrading to Release 7.x, the organizational hierarchy (even if there is only one organization) must be established to maintain appropriate visibility in the views cited above.
- When EIM is used to import or update any of the hierarchies (positions, organizations, or access groups).

The standard Release 7.x configuration includes the Generate Reporting Relationships feature as a hidden button on the Position List Applet NB. You need to go through Siebel Tools configuration to expose this button.

To expose the Generate Reporting Relationships button

1. Open Siebel Tools and navigate to the applets folder.

2. Find and select the “Position List Applet NB” applet record.

3. Right click on the record and select “Edit Web Layout.”

4. Drag the “GenReportRel” button from the Controls/Columns window into one of the button placeholders in the applet layout (that is, one of the empty “x” placeholders in the blue header area of the applet layout).

5. Repeat this step for the three different modes (Base, Edit, and Edit List) in which the applet can be displayed. The easiest way to switch between the different modes is to just use the Mode drop-down that appears in the Web Controls toolbar of Siebel Tools. After you have drag-and-dropped the button into all three modes of the applet layout, close the layout editor and save your changes.

6. Recompile the applet into your existing `siebel.srf` as used by the Web client.

7. Launch the Web client or Dedicated Web client using the SRF compiled in the previous step so that the Generate Reporting Relationships button can be invoked.

NOTE: Siebel Systems recommends that you do not make this button available in the standard `siebel.srf` file used by your organization in order to preserve control over who can press this button and when it can be pressed.

To generate reporting relationships

1. If you have an active Siebel Remote environment, confer with a Siebel Systems Administrator. The Administrator should arrange for the Transaction Processor to be paused before performing this procedure.

2. Choose Group Administration under Site Map and navigate to the Positions view in the Siebel Web client application. Click the “Generate Reporting Relationships” button in the Position List Applet NB. Please note that generating the reporting relationship may cause a large number of Siebel Remote transactions to be generated.

3. When this has completed, restart the Transaction Processor.

Setting Visibility Modes for Access Control

NOTE: This step applies primarily to development environment upgrades.

Changes regarding visibility settings are saved and embedded in an exported repository, so this step is not necessary after a production environment upgrade if you already completed a development environment upgrade. Instead, you should review and verify your visibility properties.

Before you deploy the upgraded configuration in a test or production environment, you need to make a decision about what type of visibility to use on certain business components, views, and applet picklists. This requires some evaluation. Certain areas of the Release 7.x default configuration use Catalog visibility.

- If you are prepared to define the catalog, category, and access group structure in parallel with the postupgrade development and testing phase before production rollout, then changes to visibility properties might not be necessary. This decision is implementation-specific. This decision should be carefully evaluated by your entire internal implementation team.
- If you wish to continue to use Organization visibility, you must make changes to the business component, view, and picklist properties in order to use Organization visibility.
    - Business Components with Catalog Visibility
    - Applet picklists with Auto Query Mode or New Query set to `None`.
    - Views with Catalog Visibility

After the upgrade, the following business components, views, and picklists—all of which use catalog visibility as the default configuration in Release 7.x—need to be carefully evaluated if they are used in your implementation.

Business Components

The following business components use Catalog as the default popup visibility type:

- Banter Resolution Item
- Competitor

- Component Product
- Contact Decision Issue
- Decision Issue
- ERM Internal News Category
- Internal Product
- My Competitor
- My Internal Product
- Recommended Product
- Related Issue
- Sales Tool
- Solution

To change the visibility properties

1. Log into Siebel Tools as a valid user.
2. Choose the “Types” tab.
3. Choose the Business Component object in Object Explorer.
4. Scroll across the list applet to find the property “Popup Visibility Type” and change this value from Catalog to Organization or to another valid property.

Picklists That Start in Query Mode

The following picklists, by default, start with Auto Query Mode set to `New Query` or `None`:

- Asset Mgmt - Internal Product Pick Applet
- CPG Internal Product Pick Applet
- Catalog Admin Category Assoc Applet—Competitor
- Catalog Admin Category Assoc Applet—Decision Issue
- Catalog Admin Category Assoc Applet—Literature

- Catalog Admin Category Assoc Applet—Product
- Catalog Admin Category Assoc Applet—Resolution Item
- Catalog Admin Category Assoc Applet—Solution
- Catalog Admin Category Assoc Applet—Training
- Catalog Admin Category Assoc Applet—eEvents Event
- FS Parts & Tools Pick Applet
- FS Use Plan Products Pick Applet
- Incentive Compensation Product Pick Applet
- Internal Product Pick Applet (eSales)
- Issue Assoc Applet
- Opportunity Management—Product Number Pick Applet
- Opportunity Management—Product Pick Applet
- Order Entry—Line Item Product Pick Applet
- Partner Finder List Applet
- Product Number Pick Applet
- Product Pick Applet
- Product Pick Applet—No Clear
- Product Pick Applet—No Insert
- SR Internal Product Pick Applet
- Sales Tool Pick Applet
- Service Locator List Applet
- Solution Create List Applet
- Training Course Product Pick Applet

To change the visibility properties

1. Log into Siebel Tools as a valid user.
2. Choose the “Types” tab.
3. Choose the Applet object in Object Explorer.
4. Scroll across the list applet to find the property “Auto Query Mode” and change this value from `New Query` or `None` to no value.

   To change the Auto Query Mode property to have no value, remove any search specification that would otherwise be inherited from the Business Component.

Views

The following views use Catalog as the default visibility applet type:

- Auction Place Bid
- Competitive Company Across Catalogs
- Decision Issue View
- Products across Catalogs
- SHP Sales Product View
- SWLS eChannel Solution Display View
- Sales Tools Across Catalogs
- Service Solution List View (SCW)
- Solutions Across Catalogs
- eAuction Auction Item Search View

To change the visibility properties

1. Log into Siebel Tools as a valid user.
2. Choose the “Types” tab.
3. Choose the View object in Object Explorer.

4 Scroll across the list applet to find the property “Visibility Applet Type” and change this value from `Catalog` to `Organization` or to another valid property.

For analysis and recommendations for choosing and implementing access control strategies from an upgrade perspective, please refer to Siebel SupportWeb for the *Access Control Upgrade and Migration Guide* technical note.

Resolving Duplicate EIM Mappings

CAUTION: This procedure only applies to development environment upgrades. For production environment upgrades, skip this procedure and proceed to “Producing a New Custom Configuration File.”

Any custom Enterprise Integration Manager (EIM) mappings that were added to previous versions of Siebel eBusiness Applications will be automatically migrated to the new custom repository. If custom mappings are duplicated by new mappings added as part of Siebel version 7.x, errors will result when you use EIM.

NOTE: Custom EIM mappings are only added to old _IF tables, not to new EIM tables. If you need to use the new EIM tables, you must manually add the customization to them.

The Siebel Software Configuration Utility that you ran in “Upgrading the Siebel Database Schema” on page 147 generated a file called `mapclash.txt`. This file lists any custom EIM mappings that conflict with Siebel Systems-provided mappings. The file contains the columns illustrated and defined in Table 22.

Table 22. Mapclash.txt File Column Names and Definitions

Column Name	Definition
`IT_Name`	Interface table name (example: `S_ACCOUNT_IF`)
`DT_Name`	Destination table (base table) name
`DC_Name`	Destination column (base table column) name

Before proceeding with the upgrade, use Siebel Tools as described below to delete any custom mappings listed in this file.

To delete duplicate custom EIM mappings

1. Launch Siebel Tools and select New Customer Repository.
2. From the Object Explorer, choose EIM Interface Table > EIM Table Mapping.
3. From the EIM Table applet, select the interface table name (`IT_Name`), for example, `S_ACCOUNT_IF,` and query for the interface table.
4. From the EIM Table Mappings applet, choose the name assigned for the destination table (`DT_Name`); for example, `S_ORG_EXT_X`.
5. From the Object Explorer, choose EIM Table Mapping > Attribute Mapping.

   The fourth column in this applet is the `DC_Name`.
6. Delete the duplicate mappings.

Safeguarding the New Custom Repository Export File

CAUTION: This procedure only applies to development environment upgrades. For production environment upgrades, skip this procedure and proceed to "Producing a New Custom Configuration File."

The Siebel Software Configuration Utility exports the new customized repository to a file called `custrep.dat`, located in the *`DBSRVR_PLTFRM_ROOT`* directory on the Siebel Server from which you ran the upgrade. Locate this file and make a backup copy of it. You will need it for the upgrade of the production database.

If you make *any* changes to your custom repository after running the development upgrade Siebel Software Configuration Utility, including reapplying custom extensions or modifying EIM mappings in the two previous steps, you must export a new copy of the repository before performing the production upgrade. Use the script *`DBSRVR_PLTFRM_ROOT`*`/master_exprep.ucf` to export your current repository to the file that you designated as the output file in the Siebel software configuration utility. You can use the Siebel Software Configuration Utility to invoke the `master_exprep.ucf` script which will then export your current repository.

The `master_exprep.ucf` calls the `repimexp` utility, which connects to two separate repositories—the repository whose structure is to be extracted, and the repository whose content is to be extracted. Apart from unusual circumstances, you will always extract *both* the structure and content of the same repository.

For information on exporting repositories, see the “Repository Management Tools” chapter of *Siebel Tools Reference*.

NOTE: Before you export the repository, you should make sure that all Siebel Tools projects are unlocked to promote a stable environment.

Producing a New Custom Configuration File

NOTE: You do not need to perform this procedure if you are performing a production environment upgrade and you have a development environment, because you already performed it (producing a new custom configuration file, custrep.dat) for the development environment. However, you must perform this procedure if you are performing a production environment upgrade and are upgrading *without* a development environment.

The repository merge process you performed during the development environment upgrade merged your configuration changes into a new custom configuration repository. You must now compile a new Siebel repository file (`.srf`) from this repository.

NOTE: If you intend to deploy your Siebel repository file (SRF) in more than one language, set the Tools Language setting and compile an SRF for each language. For information about how to compile an SRF for a specific language, see *Global Deployment Guide*.

To compile a new .srf file

1. Using a new Siebel Tools development workstation, connect to the database against which you ran the merge, and then open your new customized repository by choosing File > Open Repository.
2. Select “Siebel Repository.”
3. Choose Tools > Compile Project.
4. Verify or add the following information in the Object Compiler dialog box:
   - Select All Projects.
   - The name of the output Siebel Repository (`.srf`) file (the name for your custom configuration `.srf` file).
5. Click Compile.

After the compilation process has finished, distribute the new `.srf` file to each of your development Siebel Client workstations. For information about how to distribute a new `.srf` file, see *Siebel Anywhere Administration Guide*.

Regenerating the Database Template File

CAUTION: This procedure only applies to development environment upgrades. For production environment upgrades, skip this procedure and proceed to "Validating Dock Objects and Rule Definitions."

Following the upgrade, you must regenerate the SQL Anywhere template database file used by Siebel Remote. This process updates its schema to the same version as the database server. You will use the Generate New Database component from a new Siebel Server to do this.

For instructions on regenerating the SQL Anywhere Template file, refer to *Siebel Remote and Replication Manager Administration Guide*.

Extracting Developers or Siebel Tools Clients

CAUTION: This procedure only applies to development environment upgrades. For production environment upgrades, skip this procedure and proceed to "Validating Dock Objects and Rule Definitions."

Extract all Siebel Tools developers and clients using the Database Extract component from a new Siebel Server.

After you have extracted and initialized the mobile databases of all Siebel Tools developers, they must check out a read-only copy of all projects into the local database. For further instructions on setting up Siebel Tools developers, see *Siebel Tools Reference*.

Testing the Upgraded Configuration

CAUTION: This procedure only applies to development environment upgrades. For production environment upgrades, skip this procedure and proceed to "Validating Dock Objects and Rule Definitions."

You must test the upgraded configuration thoroughly to make sure that all functionality works as expected. You need to execute your full development or acceptance test plan, including at least one synchronization session from a mobile client, before proceeding to upgrade your production environment.

If you make further changes to your customized configuration, remember to re–export the repository to the `custrep.dat` file before running the production environment upgrade.

NOTE: Save the log files generated during the repository upgrade and schema upgrade. Locate the custrep.dat file in the `DBSRVR_ROOT\COMMON` subdirectory. Save the `custrep.dat` file for your production upgrade.

Validating Dock Objects and Rule Definitions

If you deploy Siebel eBusiness Applications to mobile users with local databases, you can run the `DICTUTL` utility to verify that all dock objects and rule definitions are correct. Dock objects allow mobile users to synchronize their local databases with the Siebel Server. Rules determine which data users synchronize. For more information about dock objects and rules, see *Siebel Tools Online Help* and *Siebel Remote and Replication Manager Administration Guide*.

To verify that all dock object and rule definitions are correct

1 Type the following command from `$SIEBEL_ROOT/bin` using the parameters specified in Table 23:

```
dictutl /C ODBC_DATASOURCE /U USERNAME /P PASSWORD /D
TABLEOWNER /N "REPOSITORY_NAME" /A y 2> logfile.log
```

Table 23. Command Line Flags for DICTUTL

Flag	Parameter	Description	Required
/C	*ODBC_DATASOURCE*	ODBC datasource name	Yes
/U	*USERNAME*	User name to log in to database	Yes
/P	*PASSWORD*	User password to log in to database	Yes
/D	*TABLEOWNER*	User name of tableowner	Yes

Table 23. Command Line Flags for DICTUTL

Flag	Parameter	Description	Required
/N	"*REPOSITORY_NAME*"	Name of repository for dictionary (the parameter must be bounded within double quotes)	Yes
/A	*y* or *n*	Enter the y parameter to ignore dictionary cache. Enter n if you do not want to ignore the dictionary cache.	Yes

2 Review the *LOGFILE*.log file.

Run Statistics

If your RDBMS is DB2 UDB, you should run full statistics on all tables used in your environment.

- Run `updatestats` from *`$SIEBEL_ROOT`*`/DB2UDB`.

Reorganize the P1 Index for DB2 UDB

After your upgrade to Release 7.x, you need to verify that the table S_EVT_ACT is reorganized on the P1 index. This happens because the append mode flag is set in the repository. If S_EVT_ACT is not reorganized by the P1 index, Siebel Remote performance will be negatively affected.

Recreate Server Component Definitions

After you complete your upgrade, you need to recreate server component definitions. For more information, see *Siebel Server Administration Guide*.

Reset Database Server Configuration Parameters

After you complete your upgrade, you need to reset your database server configuration to installation settings. See *Siebel Server Installation Guide* for the operating system you are using.

Clean Up Your Repository Files

CAUTION: This procedure only applies to development environment upgrades. For production environment upgrades, skip this procedure and proceed to "Additional Postupgrade Tasks for Specific Applications."

After you resolve all merge conflicts and errors, export your Prior Customer Repository as a backup for your production environment upgrade. You can use Siebel Tools to safely delete the following repositories:

- Prior standard repository (version of 5.x, 6.x. or 7.0.x)
- New standard repository (version 7.5)

Additional Postupgrade Tasks for Specific Applications

This section describes additional postupgrade tasks and considerations for the following applications:

- Call Center
- Enterprise Application Integration (eAI)
- Siebel Financial Services
- Siebel Life Sciences
- Marketing
- Siebel Purchase Orders
- Quotes
- Siebel Workflow
- Siebel eConfigurator

Call Center

NOTE: If you are upgrading from Release 7.0.x to Release 7.5, you do not need to perform this procedure.

During the upgrade to Release 7.x, employee and contact data were moved to the same tables. As a result of this change, your database will contain duplicate logins for Contact user login names.

In order to prevent the creation of duplicate logins which may occur due to this change, the Release 7.x upgrade will append the `ROW_ID` to duplicate logins. You need to resolve your user logins after the upgrade, or users may be unable to log in. For example, contact user login names will appear concatenated with their row ID.

To locate user logins that require resolution of duplicates

1. Open your Call Center application and navigate to Site Map > User Administration.

2 For each User Administration view (Employees, Persons, and Users) query the User ID field for login name=*+*. This query will bring up all names that are appended with “+<ROW_ID>”.

3 Repeat this procedure for each User Administration view (Employees, Persons, and Users).

Enterprise Application Integration (eAI)

CAUTION: This procedure only applies to development environment upgrades. For production environment upgrades, skip this procedure and proceed to “Siebel Financial Services.”

If you use Enterprise Application Integration (EAI), perform the following procedure to update the definitions of the Business Objects to account for changes in data type, length, edit format or other properties during upgrading to a new version of Siebel Applications.

To upgrade integration objects

1 Synchronize the integration objects.

2 Validate the integration objects.

3 If you receive validation errors, inactive the user keys or fields that cause the error.

4 If you receive the error “List Of” in the XML Parent Element, manually remove the value “List Of” from the XML Parent Element.

Siebel Financial Services

NOTE: If you are upgrading from Release 7.0.x to Release 7.5, you do not need to perform these procedures.

If you are upgrading a Siebel Financial Services application, perform the following tasks in addition to the postupgrade tasks described earlier in this chapter.

NOTE: The information in this section is specific to Siebel Financial Services. If you are not upgrading a Siebel Financial Services application, you should skip "Siebel Financial Services" on page 235.

Updating File System Attachments for Call Reports

When you performed the Siebel database upgrade to Release 7.x, call report attachments from `S_ACTIVITY_ATT` migrated to `S_COMM_ATT`. This data migration updated the records in the database and the pointer to the physical file, but did not update file system attachments, which use a naming convention that includes the name of the table they are associated with: *`TableName_RowID_Revision#`*`.saf`.

Therefore, you need to run a script to copy and rename all files named `S_ACTIVITY*.SAF` to `S_COMM*.SAF` so that they correspond to new table names.

To generate the file attachment update script

- From your `$SIEBEL_ROOT/bin` directory, type the command appropriate for your operating system:
  - **UNIX**

```
odbcsql /s "$ODBCDataSource" /u $TableOwner
/p $Password /separator / /a /c rem
/o $FileSystemLocation/mv_act_to_comm_att_unix.ksh
/l $LogFileLocation/mv_act_to_comm_att_unix.log
$ScriptLocation/mv_act_to_comm_att_unix.sql /v
```

where:

"ODBCDataSource" = Data source of the database (entered in quotation marks)

Tableowner = Tableowner

Password = Tableowner password

FileSystemLocation = Location of the file system

LogFileLocation = Location of the log file

ScriptLocation = Location of the script

To update the file names of call report attachments

1. Change your directory to the file system.
2. Type the command appropriate for your operating system:
   - **UNIX**

     ```
     mv_act_to_comm_att_unix.ksh
     ```

Updating File System for Household Notes and Attachments

If you are migrating your implementation to the new household design, perform the following procedure to update the file system for household notes and attachments.

The Siebel database upgrade to Release 7.x migrated household note attachments from S_ORGGRP_ATT to S_CONTACT_ATT. This data migration updated the records in the database, but did not update file system attachments, which use a naming convention that includes the name of the table they are associated with: *TableName_RowID_Revision*#.saf.

Therefore, you need to run two scripts to complete this migration. First, run a script to migrate attachments from `S_ORGGRP_ATT` to `S_CONTACT_ATT`, and notes from `S_NOTE_ORGGROUP` to `S_NOTE_CON` in the database. Then run a script to copy and rename all files named `S_ORGGRP*.SAF` to `S_CONTACT*.SAF` so that they correspond to new table names.

To update file attachments

1. Run `HHMignot.sql` to migrate attachments from `S_ORGGRP_ATT` to `S_CONTACT_ATT`.
2. From *`SIEBSRVR_ROOT/bin`*, enter the following command:

   ```
   chng_file_sys.ksh -s Source_Table -t Target_Table
   -f "File_System"
   ```

   where:

   ```
   Source_Table = Name of the original table

   Target_Table = Name of the new table to which the original data
   was migrated

   "File_System" = Name of the directory where the file system
   resides (entered inside quotation marks)
   ```

   for example,

   ```
   chng_file_sys.ksh -s S_ORGGRP_ATT -t S_CONTACT_ATT
   -f "D:\DBSRVR_ROOT\Files"
   ```

3. Review the renamed files carefully to verify that they can be accessed by Siebel Financial Services.

Opportunity Product Migration

During the upgrade, data is migrated from S_OPTY_PROD to S_REVN. As a result of this migration, if you have custom objects which point to S_OPTY_PROD, you need to remap the affected base or extension tables. Extension columns pointing to S_OPTY_PROD may need to be repointed as well.

The following base or extension tables are affected in the migration from S_OPTY_PROD to S_REVN:

- S_FN_OFFR_COLT
- S_FN_OFFR_FEE
- S_FN_OFFR_SCHD
- S_OPTYPRD_ORG
- S_OPTY_PROD1_FNX
- S_OPTY_PROD_FNX
- S_OPTY_PROD_FNXM

For example, if a child object pointed to S_OPTY_PROD, that child object needs to be manually remapped to S_REVN.

A report generated during the upgrade identifies which columns or tables you need to examine. This report, `xtndobstbl.log`, lists extension columns that reside on obsolete tables, and therefore need to be moved to alternate tables. For more information about reapplying custom extension columns on obsolete tables, see "Reapplying Custom Extensions and FK Columns on Obsolete Tables" on page 213.

Siebel Life Sciences

CAUTION: This procedure only applies to production environment upgrades. For development environment upgrades, skip this procedure and proceed to "Marketing."

If you successfully completed a production environment upgrade from Release 6.3 Life Sciences to Release 7.5, you need to rename your cfg file before you upgrade your mobile and connected clients as instructed in *Siebel Anywhere Administration Guide*. This action will prevent an error from occurring when you log in to the upgraded Release 7.5 ePharma client.

The Release 6.3 cfg file was called pharma.cfg, but in Release 7.5 the cfg file is called ePharma.cfg.

To rename your ePharma configuration file

- After you install the upgraded Client Executables and Client Executables ENU, but before you create the packager kit with Siebel Anywhere, copy the file named `ePharma.cfg` and rename the copy to `Pharma.cfg` in the new build client before you create the packager kit.

NOTE: If you do not rename the cfg file, you may receive the following error message after the client upgrade, after you install the upgraded Client Executables and Client Executables ENU, when you try to log in to the upgraded 7.5 ePharma client: `Unable to load configuration file "D:\63\client\Bin\ENU\Pharma.cfg".`

Marketing

NOTE: If you are upgrading from Release 7.0.x to Release 7.5, you do not need to perform this procedure.

Customers who use Siebel Marketing and the Marketing Server will have to manually upgrade their meta data for those cases where they have mapped to tables or fields in the Siebel OLTP or the eBusiness Data Warehouse that have changed in Release 7.0. Due to the extensive changes to the Data Warehouse schema in Release 7.0, entirely new mappings may be required. Please see the *Siebel Marketing Guide* for information on table and field meta data mappings.

Siebel Purchase Orders

CAUTION: This procedure only applies to production environment upgrades. For development environment upgrades, skip this procedure and proceed to "Quotes."

After you have upgraded to Release 7.5, you need to run a utility to update the Transaction Amount field in Payment Lines that have Purchase Order as the payment method. The utility requires the .srf that you compiled after upgrading your development environment.

The utility performs the following steps:

1. Creates a new Order Entry business object.

2. Creates an Order Entry business component and a Payments business component.
3. Checks all order records and looks at the Payment Method of corresponding Payment Lines.
4. If the Payment Method of a Payment Line is Purchase Order, it updates the Transaction Amount field to Order Total.

The utility requires the following parameters:

- `username.` Siebel user login name
- `password.` Siebel login password
- `lang.` Language used
- `cfg file.` The configuration file used to launch the application
- `data source.` Date source used from the .cfg file

To run the utility from the command line

- Enter the following:

```
Pmntupgd /u username /p password /l lang /c cfg file /d data source
```

Quotes

NOTE: If you are upgrading from Release 7.0.x to Release 7.5, you do not need to perform this procedure.

After upgrading to Release 7.x, the administrator needs to verify that the new customizable products work as needed. In order to do so, the administrator needs to validate and release the customizable product for use. This action is performed in the Product Administration screen. If the administrator does not release the product after the upgrade, customizable products will not be active in Release 7.x. Refer to *Siebel Interactive Selling Applications Upgrade Guide* for more details.

Packaged products do not require additional modifications in the postupgrade process.

After upgrade, you need to run the following statement to fix an issue where quote items had trailing spaces added accidentally. Running this script is mandatory.

```
update S_QUOTE_ITEM

setROW_ID = rtrim(ROW_ID)

,   ROOT_QUOTE_ITEM_ID = rtrim(ROOT_QUOTE_ITEM_ID)

,   PAR_SQ_ITEM_ID = rtrim(PAR_SQ_ITEM_ID)

,   PORT_VALID_PROD_ID = rtrim(PORT_VALID_PROD_ID)

,   PROD_PORT_ID = rtrim(PROD_PORT_ID)

,   INTEGRATION_ID = rtrim(INTEGRATION_ID)

;

commit

;
```

Siebel Workflow

Perform the following postupgrade steps after a successful upgrade of Siebel Workflow to Release 7.5.

Updating Inbound Workflows

- Change inbound workflows that contain a “String” type process property to pass the value into type `Binary`; otherwise, the workflow will give the following error message:

  ```
  Output argument '<Value>' in step 'Read from File' contains data
  that cannot be passed to string type property 'InputXML'. Data
  type: 'MEMBLOCK'; String representation of data body: '<?xml
  version="1.0" encoding="UTF-8"?><?'
  ```

- After an upgrade from Release 6.x to 7.5, manually change the name of the “EAI MQSeries Transport” business service to the name “EAI MQ Series Server Transport”; otherwise, the workflow will give the following error message:

  ```
  [1] Unable to create the Business Service 'EAI MQSeries Transport'
  [2] Could not find 'Class' named 'EAI MQSeries Transport'. This
  object is inactive or nonexistent.
  ```

Updating Seeded Workflows

NOTE: If you are upgrading from Release 5.x or Release 6.x, you do not need to perform this procedure.

During the upgrade from Release 7.0.x to Release 7.5, Release 7.0.x seeded workflows were replaced by Release 7.5 seeded workflows. Customizations to seeded workflows were saved and migrated, but you need to manually reimplement them in order for them to work properly. Nonseeded workflows were migrated as such, carrying over respective version numbers.

Seeded workflows have inactive status and version 0. They cannot be modified.

To use a seeded workflow

- Revise and activate the workflow so that a new copy of the seeded process will be created with the latest version number.

To incorporate prior customizations in the new seeded workflows

- Perform a manual three-way merge. That is, revise the seeded workflow (shipped as version 0), manually merge in your customizations, and then activate the workflow.

For additional eSales upgrade considerations, see *Siebel Interactive Selling Applications Upgrade Guide*.

Siebel eConfigurator

NOTE: If you are upgrading from Release 7.0.x to Release 7.5, you do not need to perform this procedure.

When the upgrade process is finished, customers will need to examine their upgraded models, in the form of customizable products, and release the products for end user use. Refer to *Siebel Interactive Selling Applications Upgrade Guide* for more postupgrade review procedures.

Upgrading to RC2 Encryption

CAUTION: If you have data that was encrypted using the Release 6.x or 7.0.x standard encryptor, and you need to be able to read this encrypted data in Release 7.5, you must upgrade your encryption method after successfully upgrading but before deploying the application.

Release 7.5 delivers a new default encryption method that is based on the RC2 standard. The previous default encryption method (called the standard encryptor) is no longer supported, and data that used the standard encryptor will not be read properly by 7.5 applications unless you upgrade your encryption method to RC2. Use the Encryption Upgrade Utility to convert unencrypted data and data that was encrypted using the standard encryptor to the RC2 encryption method.

CAUTION: Using a non-RC2 encryption method in a Unicode environment will result in irrecoverable data loss.

Perform the following procedures to upgrade your encryption method:

- Verify that all prerequisites are met. See "Prerequisites" on page 245.
- Make sure that the input file includes every column that you want to upgrade. See "Modifying the Input File" on page 245.
- If you customized business component fields to use the old encryption method, verify that you have the correct user property definitions. See "Changing User Properties" on page 246.
- Run `keydbmgr.exe` to change the password or add a new key to the database. See "Changing the Password or Adding a New Key to a Database" on page 246.
- Determine which encryption you are going to use: 128-bit or 56-bit encryption.
  - If you are upgrading to 128-bit encryption, follow the instructions for upgrading to 128-bit encryption, "To upgrade to 128-bit encryption (for the Strong Encryption Pack)" on page 248 and then continue with the upgrade to 56-bit encryption.
  - If you are upgrading to 56-bit encryption, see "To upgrade to 56-bit encryption" on page 248.

Prerequisites

In order to upgrade to the RC2 encryption method, the following prerequisites must be fulfilled:

- The Siebel Gateway and Siebel Server are installed.
- The repository has been upgraded to the Release 7.5 schema, so that a new column has been created to store the key index for the encrypted column.
- If you created or customized columns to use the old encryption method (the standard encryptor), for each encrypted column that you want to upgrade, you need to create a new column to store the key index.
- The key database (`keyfile.bin`) must already exist. (A default keyfile was created in the *`SIEBSRVR_ROOT`*`/siebsrvr/admin` directory when you installed the Siebel Server.)
- The password must be stored in the database.

Modifying the Input File

The input file `encrypt_colums.inp` indicates the table and column that store the encrypted data, and the table and column that store the key index. The input file is located in *`DBSRVR_ROOT`*`/DBSRVR/bin` directory. If you wish to execute the utility from the command line, place this file in the *`SIEBSRVR_ROOT`*`/bin` directory.

The input file must include every column that you want to upgrade. The first line of the input file indicates a table name with brackets around it. The table name should be followed on subsequent lines by all the columns that will be upgraded for that table. Each column requires a table column to store the key index, so this is specified after the column name; for example:

```
[TABLE_NAME]
COLUMN_NAME TABLE_NAME_FOR_KEY COLUMN_NAME_FOR_KEY
```

After each table, skip a line, and continue with subsequent tables. Here is a sample input file:

```
[S_ORDER]
CC_NUMBER S_ORDER CCNUM_ENCRPKEY_REF

[S_DOC_ORDER]
CC_NUMBER S_DOC_ORDER CCNUM_ENCRPKEY_REF
```

```
[S_PER_PAY_PRFL]
PAY_ACCNT_NUM S_PER_PAY_PRFL CCNUM_ENCRPKEY_REF
```

To support upgrade of non-encrypted field to RC2 encryption, add the letter `N` to the end of the column; for example:

```
[S_NEW_TABLE]
NAME S_NEW_TABLE NAME_KEY_INDEX N
```

Changing User Properties

If you customized business component fields to use the old encryption method (the standard encryptor), make sure that your custom buscomp field user properties are defined with the values provided in the table below. (An example is provided for the Quote business component.)

NOTE: By default, data encrypted using the old encryption method will use `ROW_ID` as the Encrypt Key Field. You may need to create a calculated field on each business component for the Encrypt Read Only Field.

Field User Property	**Value**	Example for the Quote BusComp
Encrypted	`Y`	`Y`
Encrypt Service Name	`RC2 Encryptor`	`RC2 Encryptor`
Encrypt Key Field	*`Key Index Field`*	`Credit Card Number Key Index`
Encrypt Read Only Field	*`Read Only Field`*	`Credit Card Number Read Only`

Changing the Password or Adding a New Key to a Database

If you need to change the password or add a new key to your database, perform the following steps:

To change the password or add a new key

- Run `keydbmgr.exe` to change the password or add a new key to your database. From *$SIEBSRVR_ROOT/bin,* enter the following command:

  ```
  keydbmgr.exe /U USERNAME /P PASSWORD /C
  CONFIGURATION_FILE_NAME /L LANGUAGE
  ```

  where:

  - *USERNAME* = user name
  - *PASSWORD* = password
  - *CONFIGURATION_FILE_NAME* = name of the configuration file (the default is `siebel.cfg`). Make sure that the configuration file is pointing to the correct database.
  - *LANGUAGE* = The base language for your installation (the default is `enu`)

  for example,

  ```
  keydbmgr.exe /u sadmin /p PASSWORD /c siebel.cfg /l enu
  ```

  NOTE: Use the default password `kdbpass` to log in and change the password.

Upgrading Your Encryption Method

Follow the instructions below for the encryption you are going to use: 56-bit encryption or 128-bit encryption.

- If you purchased the Strong Encryption Pack for 128-bit encryption, follow the instructions, "To upgrade to 128-bit encryption (for the Strong Encryption Pack)" on page 248, and then continue with the upgrade to 56-bit encryption.

  NOTE: Verify encryption requirements and constraints for your deployment before you upgrade your encryption to 128-bit.

- If you will perform the standard encryption upgrade to 56-bit encryption, skip the upgrade to 128-bit encryption, and go directly to the instructions "To upgrade to 56-bit encryption" on page 248.

To upgrade to 128-bit encryption (for the Strong Encryption Pack)

1 Backup your existing keyfile.

2 Run the keydbmgr.exe utility to change the keyfile password. See “” on page 246.

3 Install the Strong Encryption Pack that you purchased separately.

4 Run `keydbupgrade.exe` to upgrade to 128-bit encryption.
From *$SIEBSRVR_ROOT/bin,* enter the following command:

```
keydbupgrade.exe /U USERNAME /P PASSWORD /C
CONFIGURATION_FILE_NAME /L LANGUAGE
```

where:

- *USERNAME* = user name
- *PASSWORD* = password
- *CONFIGURATION_FILE_NAME* = name of the configuration file (the default is `siebel.cfg`)
- *LANGUAGE* = The base language for your installation (the default is `enu`)

for example,

```
keydbupgrade.exe /u sadmin /p PASSWORD /c siebel.cfg /l enu
```

5 Continue by upgrading to 56-bit encryption. See “To upgrade to 56-bit encryption.”

6 Use `srvrmgr` to update `entparam` in the Siebel Gateway.

Change the `entparam` password=*db_password*.

7 Restart the server.

8 Repeat these steps on each machine that has an existing Strong Encryption Pack installed.

To upgrade to 56-bit encryption

1 Verify that the input file `encrypt_colums.inp` includes all the columns that you want to upgrade. If necessary, review “Modifying the Input File” on page 245.

2 Run `encryptupg.exe` to upgrade to 56-bit encryption.
From *`$SIEBSRVR_ROOT/bin,`* enter the following command:

```
encryptupg.exe /U USERNAME /P PASSWORD /C ODBC_CONNECT_STRING
/D TABLEOWNER /J INPUT_FILE /N REPOSITORY_NAME /K KEY_FILE_NAME
/L LOG_FILE
```

where:

- *`SIEBSRVR_ROOT`* = root directory of your Siebel Server installation
- *`USERNAME`* = user name for the database
- *`PASSWORD`* = password for the database
- *`ODBC_CONNECT_STRING`* = ODBC connect string for the database
- *`TABLEOWNER`* = tableowner for the database
- *`INPUT_FILE`* = the name of your input file (the default is `encrypt_columns.inp`)
- *`REPOSITORY_NAME`* = repository name (the default is `Siebel Repository`)
- *`KEY_FILE_NAME`* = the absolute path to the key file (the default location is `SIEBSRVR\admin\keyfile.bin`).
- *`LOG_FILE`* = the default log file is `encryptupg.log`

NOTE: If you have custom encrypted fields, you can validate that all business component fields are pointing to the same column by appending the command with "`/v y`". For example,

```
encryptupg.exe /U USERNAME /P PASSWORD /C ODBC_CONNECT_STRING /D
TABLEOWNER /J INPUT_FILE /N REPOSITORY_NAME /K KEY_FILE_NAME /L
LOG_FILE /v y
```

But, some business components may not use the same column intentionally.

3 Repeat Step 2 on page 249, `encryptupg.exe`, for each additional database. For each database, use the appropriate user name, password, ODBC connect string, and tableowner.

4 After the upgrade is complete, compile a new Siebel repository file. See "Producing a New Custom Configuration File" on page 229.

Troubleshooting

If you fail to change user properties, you may get one or more of the following repository validation error messages. Examples of possible errors and sample remedies are provided below:

- **Possible Error:** Validation failed for Personal Payment Profile buscomp, Account Number field does not have field user property Encrypt Key Field defined or activated.

  Sample Fix: Go to the Personal Payment Profile business component, Account Number field, and verify that the following field user properties exist:

  Encrypted = `Y`

  Encrypt Key Field = *`KEY_INDEX_FIELD`*

  Encrypt Service Name = `RC2 Encryptor`

- **Possible Error:** Validation failed for S_PER_PAY_PRFL table, PAY_ACCNT_NUM column. User properties for Personal Payment Profile buscomp, Account Number field does not match FS Invoice buscomp, Credit Card Number field.

  Sample Fix: Go to FS Invoice buscomp, Credit Card Number field and make sure the three required field user properties match the ones in Personal Payment Profile buscomp, Account Number field.

- **Possible Error:** Validation failed. Personal Payment Profile buscomp, Payment Key Index index field's table and column name does not match FS Invoice buscomp, Credit Card Key Index index field.

  Sample Fix: Go to the Personal Payment Profile buscomp, and make sure that the Payment Key Index field points to the same column as the Credit Card Key Index field in the FS Invoice buscomp.

Postupgrade Tasks for Global Deployments

The following procedures are for global deployments, deployments that intend to become global in the future, and deployments that wish to take advantage of global deployment features.

- Setting Up Your Environment to Support Global Time Zone
- “Migrating to Unicode” on page 253

Thoroughly read *Global Deployment Guide* before you internationalize any environment.

Setting Up Your Environment to Support Global Time Zone

Global deployments typically span multiple geographies and have users working in several different time zones. The global time zone feature converts and stores date and time data using the Universal Time Coordinated (UTC) standard, which is equivalent to Greenwich Mean Time, but without daylight savings time. This feature allows you to track dates and times in a common format across time zones.

NOTE: Although enabling your environment for global time zone is optional in Release 7.x, it is strongly recommended that you operate your production environment with global time zone enabled.

CAUTION: If you previously upgraded to Release 7.0.x and you enabled your environment for global time zone, you do not need to repeat the same procedure. Instead, you need to upgrade data from columns that were *not* UTC-enabled in Release 7.0.x that are UTC-enabled in Release 7.5.

Perform one of the following procedures, as appropriate to your upgrade path:

- **For upgrades from Release 6.x to 7.x.** See “Enabling Global Time Zone” on page 252 for information about enabling global time zone support after an upgrade.
- **For upgrades from Release 7.0.x to 7.5.** See “Upgrading UTC Delta Columns” on page 252 for information about upgrading 7.0.x data for columns that were *not* UTC-enabled in Release 7.0.x that changed to UTC-enabled in Release 7.5.

Enabling Global Time Zone

To enable global time zone support, after your upgrade is complete, you need to set the global time zone parameter (Universal Time Coordinated system preference) to `TRUE` through Siebel Tools. The UTC system preference is not enabled after any upgrade to Release 7.x. To enable the global time zone feature, you will need to run the UTC conversion utility. The high-level steps you need to perform are outlined below (see *Global Deployment Guide* for complete instructions).

To enable global time zone support after an upgrade

1. Stop Siebel Servers.
2. Set UTC system preference to `FALSE` before you convert your historical data.
3. Prepare your data for conversion to global time zone, as instructed in *Global Deployment Guide*.
4. Convert your historical data, using the UTC conversion utility, to make all existing date/time values consistent with global time zone logic. Perform this step as instructed in *Global Deployment Guide*.
5. After you have confirmed that your UTC conversion was successful, turn on global time zone by setting the UTC system preference to `TRUE`. To reset this parameter through Server Manager, navigate to Application Administration > System preferences.
6. Bring up Siebel Servers and Web Server.

Upgrading UTC Delta Columns

If, before upgrading to Release 7.5, you previously upgraded to Release 7.0.x and you already ran the UTC conversion utility, perform the following steps to upgrade the delta columns (those that were not UTC-enabled in 7.0.x but are UTC-enabled in 7.5).

CAUTION: Back up your database before you run the UTC delta conversion procedure.

To upgrade 7.0.x data for columns that have changed to UTC-enabled in 7.5

1. Open the `master_utc.ucf` file from *`DBSRVR_ROOT/DATABASE_PLATFORM/`*`master_utc.ucf`, and edit the File Name parameter by replacing `driver_utc.ucf` with `driver_utc_delta.ucf`.

2 Launch the Database Server Configuration Utility from the command prompt by typing:

```
dbsrvr_config.ksh
```

NOTE: If you are prompted to select an .scm file, select `dbsrvr.scm`.

3 When you reach the Database Server Options screen, select Run Database Utilities.

4 At the Database Utility Selection screen, select UTC Upgrader.

5 Continue until you are prompted to run the Siebel Upgrade Wizard (srvrupgwiz), then click cancel.

For more information about UTC, see *Global Deployment Guide*.

Migrating to Unicode

Verify that your encryption method has been upgraded before you proceed with the Unicode migration.

CAUTION: Use of a non-RC2 encryption method in a Unicode environment might result in irrecoverable data loss. See "Upgrading Your Encryption Method" on page 247.

After you have upgraded your Siebel eBusiness Application to Release 7.5, you can migrate your upgraded database from a non-Unicode code page (or character set) to Unicode.

NOTE: Migration to Unicode is optional and requires the assistance of Siebel Expert Services.

For information about Unicode and global deployment for Siebel eBusiness Applications, see *Global Deployment Guide*.

Planning Considerations for the Unicode Migration

If you are planning to migrate your upgraded application to Unicode, you need to consider the following points:

Migration to Unicode will increase the size of your database. For this reason, you need to allocate additional space for your database before migrating to Unicode. For more information, contact Siebel Expert Services.

Migration to Unicode may cause truncation of certain data in DB2 UDB databases. In the past, long columns with a type of varchar could have a maximum length of 16,383 characters. However, in Unicode, the maximum length of long columns with a type of varchar is 16,350. During the migration to Unicode, long columns of type varchar that exceed 16,350 will be truncated. To prevent this, you can perform tasks to identify which data may be truncated and take appropriate measures before migration. For more information, contact Siebel Expert Services.

DDLDICT does not preserve custom tablespace information for DB2 UDB databases. This will present a problem during your migration to a Unicode code page, because you need to know which tables need to be recreated. If your RDBMS is DB2 UDB and you use custom tablespaces, you need to modify the upgrade scripts to handle custom tablespaces.

NOTE: Contact Siebel Expert Services for instructions about how to modify upgrade scripts to handle custom tablespaces.

Migration to Unicode may affect integration with third party systems. For more information, contact Siebel Expert Services.

Certain handheld and wireless application platforms do not support Unicode. For more information, contact Siebel Expert Services.

Release 7.5 supports two types of Unicode:

- **UTF-8.** UTF-8 is supported for Oracle databases for Release 7.5. UTF-8 uses the same encoding for Western European languages.
  - Occupies one byte for Western European languages.
  - Occupies up to three bytes for some Asian languages, such as Japanese.
- **UCS-2.** UCS-2 is supported for DB2 UDB and Microsoft SQL Server databases for Release 7.5. UCS-2 does not map one-to-one with Western European languages, but it occupies two bytes for all languages.
  - Occupies two bytes for Western European languages.

- Occupies two bytes for Asian languages.

Migration Process

If you decide to migrate your database to Unicode, you will follow the process described here. Migrations to Unicode require the assistance of Expert Services.

To migrate to Unicode, you perform a two-step process

1. Upgrade your non-Unicode code page database to Release 7.5.

2. Migrate your upgraded Release 7.5 database from a non-Unicode code page to Unicode. To perform the migration, you can use either database vendor native utilities or Siebel utilities. For more information, contact Siebel Expert Services.

> CAUTION: You cannot use a non-Unicode code page for your development environment, and then later, migrate to Unicode for your production environment.

Upgrading Regional Servers

CAUTION: This procedure only applies to production environment upgrades. For development environments, skip this procedure and proceed to "Continuing with the Upgrade."

After the production upgrade has executed successfully and the system integrity has been adequately verified, database extracts must be run for all Regional Servers in the deployment. Before initialization of the Regional Nodes, you must drop all Siebel tables, views, and indexes using the appropriate documentation from the RDBMS vendor. Please refer to the *Siebel Remote and Replication Manager Administration Guide* for more information on database extracts, and regional server initialization.

Upgrading Mobile and Connected Clients

CAUTION: This procedure only applies to production environment upgrades.

After a successful upgrade of Siebel eBusiness Applications, you are ready to upgrade mobile and connected clients.

To upgrade Siebel client software on connected and mobile user workstations

1. Uninstall the previous version of the Siebel client application.
2. Install the Release 7.5 version of the Siebel client application. You can use either of the following methods for the installation:
   - Use the Release 7.5 installation CD.
   - Use a custom installer created using the Siebel Packager utility. For more information about the Packager utility, see *Siebel Web Client Administration Guide*.

NOTE: See *Siebel Web Client Administration Guide* for detailed instructions about how to uninstall or install Siebel clients.

Continuing with the Upgrade

To continue with the upgrade, follow the instructions below for the environment you are upgrading:

- **Development Environments.** Go to Chapter 6, “Migrating the Client Configuration to the Web Client.” If you are upgrading from Release 7.0.x, you do not need to migrate the client configuration; proceed to Chapter 3, “Preupgrade Tasks,” to begin preupgrade tasks for the production environment.

  NOTE: If you choose to migrate the Client configuration to the Web Client after upgrading your production environment, (instead of migrating the client configuration after your development environment upgrade), go to Chapter 3, “Preupgrade Tasks,” to begin preupgrade tasks for the production environment.

- **Production Environments.** Go to Chapter 6, “Migrating the Client Configuration to the Web Client.” If you completed migration of the client configuration to the Web Client as part of the development environment upgrade, you have completed post upgrade tasks and should thoroughly test your upgraded configuration.

  NOTE: You only need to migrate the client configuration to the Web Client if you are upgrading from Release 5.x or 6.x.

Migrating the Client Configuration to the Web Client 6

This chapter describes procedures you need to perform to migrate to the new Release 7.x Siebel Web Client.

To better understand which phase of the upgrade process you are in and what tasks still remain, refer to Figure 8. Figure 8 illustrates where you are in the upgrade process.

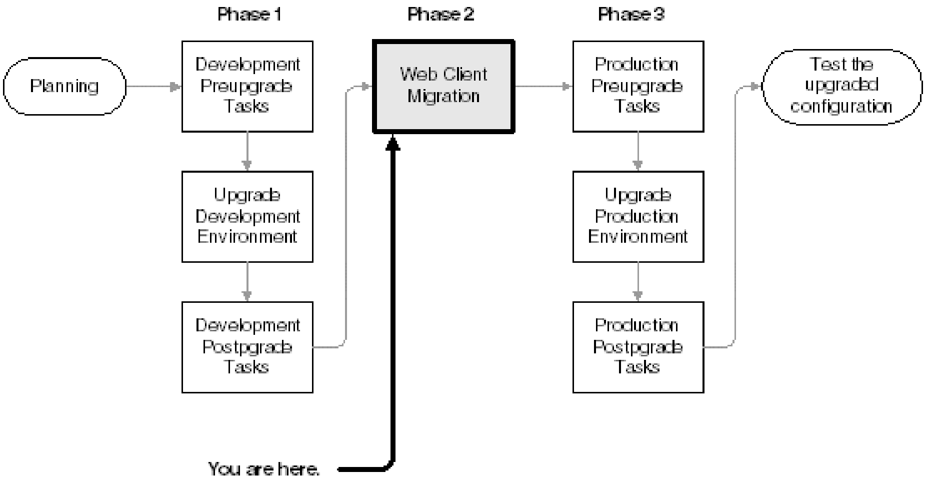

Phase 1
Phase 2
Phase 3
Planning
Development Preupgrade Tasks
Upgrade Development Environment
Development Postpgrade Tasks
Web Client Migration
Production Preupgrade Tasks
Upgrade Production Environment
Production Postpgrade Tasks
Test the upgraded configuration
You are here.

Figure 8. You Are Here

NOTE: If you are upgrading from Release 7.0.x to Release 7.5 (that is, if you previously upgraded to Release 7.0.3 or 7.0.4 and you already performed this migration), you do not need to repeat the procedures described in this chapter.

About Migrating to the Siebel Web Client

In Release 7.x, the Siebel Web Client is used as the deployment framework for all Siebel eBusiness Applications. The Siebel Web Client is a zero-footprint client that provides high levels of functionality and interactivity without requiring persistent client-side software installation. Three areas need additional attention as part of the migration to the Release 7.x Web Client. These are:

- Customized applets and views
- Scripts on UI Objects and with references to UI constructs
- Client-side interfaces to external desktop applications

The Release 7.x Application Upgrader includes a new Web Client Migration Wizard that helps to upgrade customized view and applet definitions used by the Siebel Dedicated Client, Windows Thin Client, and Java Thin Client in previous Siebel releases to the new Web Client. This wizard is run after the repository has been upgraded.

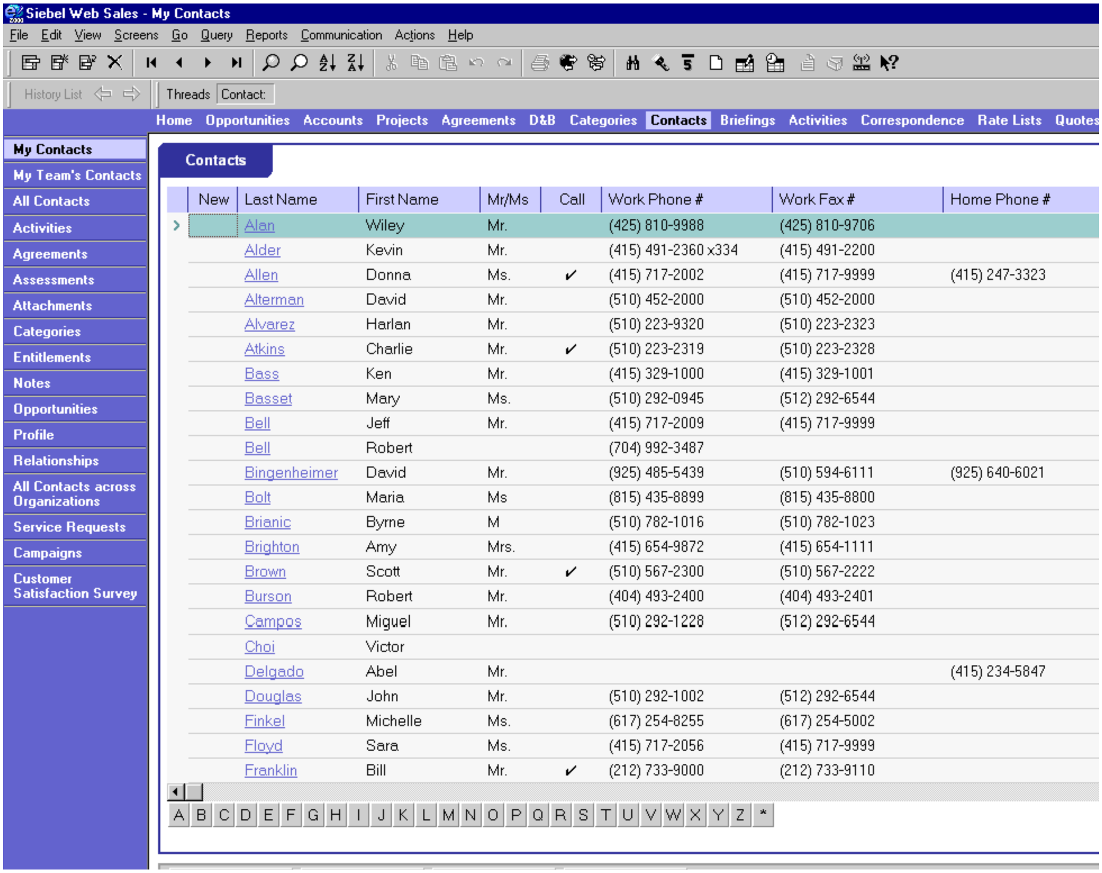
The Web Client Migration Wizard supports a batch mode that converts multiple view or applet definitions at a time, using customizable models to automatically bind the user interface controls to Web Client templates. While this process is automated, you may want to make further manual adjustments to the templates produced to make sure that your customized interface takes full advantage of the interactivity and functionality of the Release 7.x Web Client.

NOTE: This process is only required for views or applets customized in previous Siebel releases, because the Release 7.x Web Client includes standard versions of all applets and views from previous Dedicated Client releases.

The Web Client Migration Wizard will not migrate Siebel VB or Siebel eScript scripts attached to applets. These scripts will either need to be moved to the Server or to be rewritten in JavaScript, the Web Client-supported language for user interface scripting.

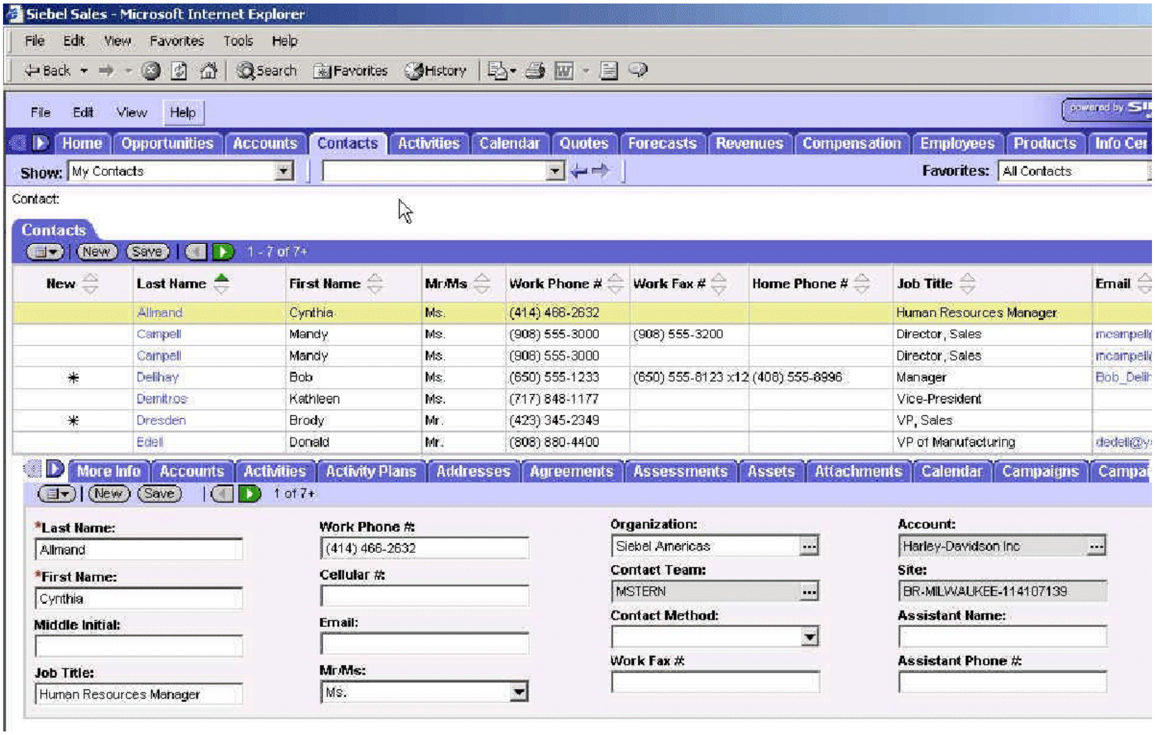
The Release 7.x Web Client supports most of the events and methods on user interface objects supported by previous Siebel releases. In addition, the Web Client provides comparable inbound and outbound client-side integration capabilities to what was available in the Business Object Interfaces in previous releases of the Siebel Dedicated Client.

The following screenshots in Figure 9 and Figure 10 on page 264 show a view that has been migrated from the Windows Client to the Web Client using the Web Client Migration Wizard.


Siebel Web Sales - My Contacts
File Edit View Screens Go Query Reports Communication Actions Help
History List
Threads
Contact:
Home Opportunities Accounts Projects Agreements D&B Categories Contacts Briefings Activities Correspondence Rate Lists Quotes
My Contacts
My Team's Contacts
All Contacts
Activities
Agreements
Assessments
Attachments
Categories
Entitlements
Notes
Opportunities
Profile
Relationships
All Contacts across Organizations
Service Requests
Campaigns
Customer Satisfaction Survey
Contacts
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z *

New	Last Name	First Name	Mr/Ms	Call	Work Phone #	Work Fax #	Home Phone #
	Alan	Wiley	Mr.		(425) 810-9988	(425) 810-9706	
	Alder	Kevin	Mr.		(415) 491-2360 x334	(415) 491-2200	
	Allen	Donna	Ms.	✓	(415) 717-2002	(415) 717-9999	(415) 247-3323
	Alterman	David	Mr.		(510) 452-2000	(510) 452-2000	
	Alvarez	Harlan	Mr.		(510) 223-9320	(510) 223-2323	
	Atkins	Charlie	Mr.	✓	(510) 223-2319	(510) 223-2328	
	Bass	Ken	Mr.		(415) 329-1000	(415) 329-1001	
	Basset	Mary	Ms.		(510) 292-0945	(512) 292-6544	
	Bell	Jeff	Mr.		(415) 717-2009	(415) 717-9999	
	Bell	Robert			(704) 992-3487		
	Bingenheimer	David	Mr.		(925) 485-5439	(510) 594-6111	(925) 640-6021
	Bolt	Maria	Ms		(815) 435-8899	(815) 435-8800	
	Brianic	Byrne	M		(510) 782-1016	(510) 782-1023	
	Brighton	Amy	Mrs.		(415) 654-9872	(415) 654-1111	
	Brown	Scott	Mr.	✓	(510) 567-2300	(510) 567-2222	
	Burson	Robert	Mr.		(404) 493-2400	(404) 493-2401	
	Campos	Miguel	Mr.		(510) 292-1228	(512) 292-6544	
	Choi	Victor					
	Delgado	Abel	Mr.				(415) 234-5847
	Douglas	John	Mr.		(510) 292-1002	(512) 292-6544	
	Finkel	Michelle	Ms.		(617) 254-8255	(617) 254-5002	
	Floyd	Sara	Ms.		(415) 717-2056	(415) 717-9999	
	Franklin	Bill	Mr.	✓	(212) 733-9000	(212) 733-9110	

Figure 9. Contact List View – Windows Client


Siebel Sales - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites History
File Edit View Help
powered by Siebel
Home Opportunities Accounts Contacts Activities Calendar Quotes Forecasts Revenues Compensation Employees Products Info Cer
Show: My Contacts
Favorites: All Contacts
Contact:
Contacts
New Save 1 - 7 of 7+
More Info Accounts Activities Activity Plans Addresses Agreements Assessments Assets Attachments Calendar Campaigns Campai
New Save 1 of 7+

New	Last Name	First Name	Mr/Ms	Work Phone #	Work Fax #	Home Phone #	Job Title	Email
	Allmand	Cynthia	Ms.	(414) 466-2632			Human Resources Manager	
	Campell	Mandy	Ms.	(908) 555-3000	(908) 555-3200		Director, Sales	mcampell
	Campell	Mandy	Ms.	(908) 555-3000			Director, Sales	mcampell
*	Delihay	Bob	Ms.	(650) 555-1233	(650) 555-8123 x12	(408) 555-8996	Manager	Bob_Delih
	Demitros	Kathleen	Ms.	(717) 848-1177			Vice-President	
*	Dresden	Brody	Mr.	(423) 345-2349			VP, Sales	
	Edell	Donald	Mr.	(808) 880-4400			VP of Manufacturing	dedell@y

*Last Name: Allmand
*First Name: Cynthia
Middle Initial:
Job Title: Human Resources Manager
Work Phone #: (414) 466-2632
Cellular #:
Email:
Mr/Ms: Ms.
Organization: Siebel Americas
Contact Team: MSTERN
Contact Method:
Work Fax #:
Account: Harley-Davidson Inc
Site: BR-MILWAUKEE-114107139
Assistant Name:
Assistant Phone #:

Figure 10. Contact List View – Web Client

The remainder of this chapter outlines the migration process and provides detailed instructions for performing migration tasks.

Migration Approach

This section provides an overview of migration tasks. Detailed migration tasks are described later in the chapter.

NOTE: If you are upgrading from Release 7.0.x to Release 7.5, (for example, from Release 7.0.3 or 7.0.4 to Release 7.5), you do not need to perform the migration described in this chapter.

Applets and Views

In Release 7.x., applets and views in the user interface are rendered using combinations of object definitions stored in the Siebel repository and layout and formatting information contained in applet Web templates. Objects such as controls and list columns are mapped to placeholders in Web templates. At run-time, the Siebel Web Engine creates these objects, places them in the appropriate spot in the Web template, retrieves relevant data from the database, and uses HTML contained in the Web template to display the applet in the user interface.

Migrating Applet and View Customizations

All the standard applets and views that are shipped as part of the Release 7.x eBusiness Applications have Web layouts. However, if you have made customizations to applets and views, you need to migrate these customizations to the Web client. Siebel provides the Web Client Migration Wizard and the Web Layout Wizard to help you perform these migrations by converting UI elements to the Release 7.x Web Client.

NOTE: Siebel Systems recommends that you use the Web Client Migration Wizard or the Web Layout Wizard to migrate your new and customized applets and views to the Web client.

The following types of applet and view customizations need to be migrated:

- New applets
- New views

- Customized applets (applets to which you have added or deleted controls or list columns)
- Customized views (views to which you have added or deleted applets)

NOTE: For detailed migration steps, see "Migrating Applets and Views" on page 271.

Migration Wizard Benefits

Using the migration wizards can benefit you and your organization in many ways. These benefits include:

- Standardization of display for applets and views
- Minimization of the number of templates required and the cost of creation and maintenance
- Simplification of application-wide modification and upgrade

Scripts

For upgrades to Release 7.x, some scripts on UI objects need to be migrated to Browser script or Server script.

Scripting in the Web Client

In Release 7.x, there are two basic types of scripts you can write to add procedural logic to an application configuration—Server scripts and Browser scripts.

Server Scripts

Server scripts execute in the Object Manager. They existed in prior releases and continue to be supported in the Release 7.x. These scripts are written in Siebel VB (for Windows platforms) and Siebel eScript (for Windows or UNIX).

The following are the various types of server scripts:

- Business Component Server Script
- Business Service Server Script

- Application Server Script
- Applet Server Script

These scripts allow developers to script the behavior of Business Components, Business Services, Applications, and Applets, by writing event handlers for the various events exposed by the scripting model.

Browser Scripts

In Release 7.x, there is a new type of script called Browser script that executes in and is interpreted by the Browser. Browser scripts are written in JavaScript and interact with the Document Object Model (DOM) as well as the Siebel Object Model available in the High Interactivity Web Client framework. A developer can script the behavior of Siebel events as well as the browser events that are exposed through the DOM. Siebel Tools allows you to write scripts by selecting the appropriate User Agent. For information about browser script techniques, see Technical Note 386 *Browser Scripting Techniques in Siebel 7*, available on Siebel Support Web. For more information about events that are exposed through the DOM, see *Siebel Tools Online Help*.

NOTE: The DOM for Internet Explorer and Netscape Navigator are different.

Scripts to Be Migrated

All 6.x scripts written on applets will need to be either migrated to Applet Server Scripts or BusComp Server Scripts, or rewritten as Browser scripts. The 6.x scripts on the Application, Business Services, and Business Components will need to be reviewed to identify references to methods, such as UI methods, which cannot be accessed from Server scripts.

NOTE: During an upgrade from Release 7.0.x (7.0.3 or 7.0.4) to Release 7.5, the 7.0.x browser scripts and objects that contain 7.0.x browser scripts will be automatically upgraded to 7.5.

Identifying Scripts to Be Reviewed or Migrated

You can identify the number of 6.x applet scripts to be reviewed or migrated by looking at the Object List Editor in Flat mode in Siebel Tools.

To help locate server side scripts that may need to be modified, in Siebel Tools you can run the Validator on BusComp, Business Service, and Applications for which the “Scripted” flag is set to “True.”

Applet Scripts

Applet script migration falls into the following major areas:

- Certain 6.x applet scripts such as Control_Click event handlers may be moved to Business Components.
- Release 6.x ActiveX Controls that have been scripted will need to be rescripted in Browser script.
- Other 6.x applet scripts need to be rewritten in Browser script.

BusComp and Application Scripts

Release 6.x BusComp and Application scripts that contain references to UI methods need to be modified. For example, calls to MsgBox need to be replaced by calls to the RaiseError/RaiseErrorText method. Other modifications may include moving scripts to Browser script. For more information, see *Siebel Tools Online Help*.

NOTE: For more information about migrating scripts, see “Migration of Scripts” on page 288.

Client-Side Interfaces

In Release 7.x, outbound and inbound COM interfaces are available. However, you may need to perform migration tasks to achieve optimal functionality.

Outbound COM Interfaces

As in previous releases, you can invoke desktop applications, such as Microsoft Excel, from the Siebel client.

In Release 7.x, this functionality is available through Browser scripts on Microsoft Internet Explorer only.

Inbound COM Interfaces

In previous releases, desktop applications communicated with the Siebel Client through an Automation interface in the Windows Client.

In Release 7.x., the High Interactivity Web Client provides similar automation interfaces.

NOTE: For more information, see "Migration of Client-Side Interfaces" on page 294.

Migration Tasks

This section covers the tasks involved in the three major areas of migration—applet and view migration, script migration, and client-side interface migration.

Exposing Hidden Properties

In Release 7.x, certain properties in Siebel Tools are no longer applicable for the Web Client. By default, these properties are hidden. Before you begin the migration to the Siebel 7.x Web Client, you should expose these properties. Exposing these properties facilitates the migration of your customizations to Release 7.x.

To expose hidden properties

1. Using a text editor, open the Tools.cfg file, located in the `SiebelToolsDir\bin\<lang> directory`.
2. Under the [Siebel] section in the Tools.cfg file, locate the parameter titled `ClientConfigurationMode`.

   The default value for `ClientConfigurationMode` is `Web`, which hides properties which are no longer applicable.
3. To expose these attributes for migration purposes, change the value from `Web` to `All`.
4. Save and close the Tools.cfg file.
5. Launch Siebel Tools and note the previously hidden properties are once again visible.

NOTE: When you have completed the migration to Release 7.x, it is recommended that you set the `ClientConfigurationMode=Web` to hide properties which are no longer applicable.

Migrating Applets and Views

To transform your customized applets and views to closely corresponding Web client applets and views, you can use either the Web Client Migration Wizard or the Web Layout Wizard. Using the wizards, you select model applets and views to base your customized applets and views on. The model applets and views you select determine the Web layouts, including the layout of controls, of your migrated applets and views.

NOTE: Make sure that the model applets you select contain a sufficient number of Web Template Items to map all items in your customized applets.

The degree of control you want to have over the migration process will determine which wizard you use.

- **Web Client Migration Wizard.** Use the Web Client Migration Wizard if you:
  - Want all your applets of a type to have one Web layout
  - Want all of your views of a type to have one Web layout
  - Have selected one model applet of each type and one model view of each type that meet your needs
  - Want to migrate your applets and views in large batches
- **Web Layout Wizard.** Use the Web Layout Wizard if you:
  - Want to use more than one model applet for each applet type
  - Want to use more than one model view for each view type
  - Want to migrate your applets and views in multiple batches

NOTE: For more information about using the Web Layout Wizard, see *Siebel Tools Reference*.

Model Applets and Views

The Siebel repository contains models for each type of applet and view. These model objects have been designed to support diverse migration needs. For example, the model form applet has the standard Web applet buttons, Web templates, and a large number of controls mapped to Web template placeholders.

Before using model applets and views, you should compare the applets and views you intend to migrate with the structure of the models to ensure that the models suit your needs. If necessary, you can modify model applets and views to meet your requirements. For example, you can modify models to use different Web templates, controls, or control mappings. Alternatively, you can use your own custom Web applets and views with the wizards. If you choose to use your own model applets and views, you should be sure that you fully understand the behavior of the wizards.

CAUTION: If you choose to use your own model applet with the wizards, you must check that the `HTML Sequence` property is defined for all controls in the applet. The `HTML Sequence` property determines the position of a control relative to other controls in an applet, indicating tab sequence. If controls in a model applet do not have `HTML Sequence` numbers defined, the wizards may fail.

You may want to test the wizards against a few applets and views before running the wizards against an entire repository. To do this, see "Remigrating Applets" on page 283 and "Remigrating Views" on page 286.

Table 24 lists the preconfigured model applets and views available in Release 7.x:

Table 24. Model Applets and Views

Model Applets and Views
Model Chart applet
Model Form applet
Model List applet
Model MVG applet
Model Pick applet
Model Tree applet

Table 24. Model Applets and Views

Model Applets and Views
Model More Applets view
Model One Applet view
Model Tree Applet view

The wizards map items from the dedicated client layout to the Web client using the following information from model applets and views:

- **Modes to map Applet Web templates.** Modes determine the kind of actions available in applets to users. For example, some applets are read-only. Other applets allow users to edit data. Typical modes include: Base, Edit, and Edit List. The mode also determines which buttons appear in Web templates. For example, the Edit button appears in applets set to Edit mode, but does not appear in applets set to Base (Read Only) mode. For more information, see *Siebel Tools Reference*.

- **Web templates to use for each mode.** Templates contain placeholders for applets and are used to render Siebel views. The mode selected determines the Web template to be used. For more information about Web templates, see *Siebel Tools Reference*.

  NOTE: For Chart and Tree applets, the wizard only uses the Web Template information. For this reason, any applet of the appropriate type may be used as a model applet for Chart and Tree applets.

- **The sequence in which textboxes are mapped.** For form applets, the wizards determine the new relative placement of controls by establishing a placement sequence based on the Top and Left properties of each control and then correlating the placement sequence to Web template item identifiers in the model applet. The process is the same for list applets, except that the wizards determine the placement sequence of the list columns based on their sequence property. For more information, see *Siebel Tools Reference*.

- **The buttons that appear on migrated applets.** Release 7.x Web applets typically contain additional buttons that were not used by their dedicated client predecessors. For example, the Next and Previous buttons used to scroll through record sets in the Web client did not appear in the dedicated client. During migration, the wizards automatically propagate buttons contained in the model applet so that these types of buttons do not require manual work.

NOTE: Siebel Systems recommends that you use the Web Client Migration Wizard to migrate customized and new applets and views.

Applet Migration Examples

The examples provided below illustrate the way in which the wizards use Web Template Items in model applets to create Web layouts that closely correspond to dedicated client applets.

Figure 11 on page 274 shows a custom form applet, the ACME Contact Form Applet, as viewed in the Siebel 2000 Applet Designer. The circled numbers show the placement sequence of each textbox as determined by the Web Client Migration wizard. Since the Top and Left properties of controls were used to determine the placement of controls within a typical Siebel 2000 applet, these properties form the basis of the migration placement sequence. The placement sequence progresses from the upper left corner to the lower right corner by moving from left to right, then top-to-bottom pattern.

Buttons that are on the applet (referenced by the circled numbers in Figure 11) are not migrated the same way as text boxes since the standard Release 7.x Web applet layout typically places these in the applet header. The circumstances in which custom buttons are automatically exposed to the migrated applet are discussed in more detail below.

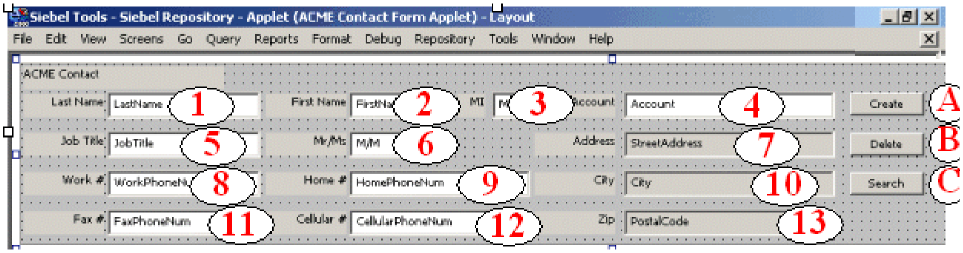
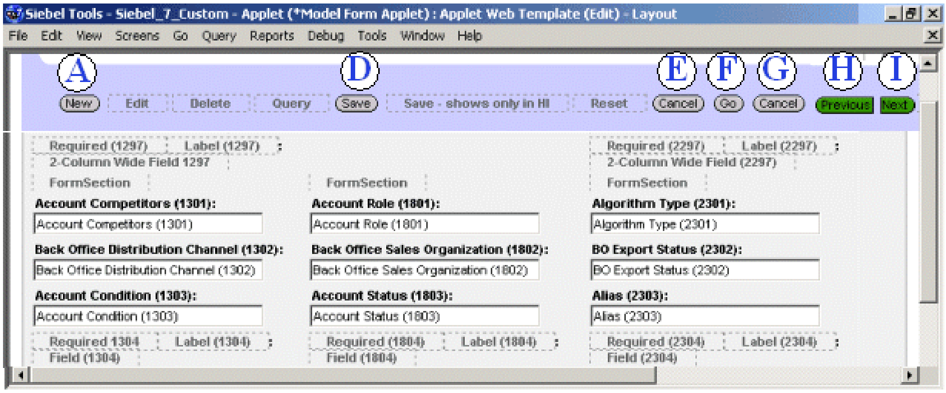


Figure 11. ACME Contact Form Applet in Siebel 2000 Applet Designer

[Figure 12](#) shows a simplified Model Form Applet in the Release 7.x Web Applet Layout Editor.


Siebel Tools - Siebel_7_Custom - Applet (*Model Form Applet) : Applet Web Template (Edit) - Layout
File Edit View Screens Go Query Reports Debug Tools Window Help
A
D
E
F
G
H
I
New
Edit
Delete
Query
Save
Save - shows only in HI
Reset
Cancel
Go
Cancel
Previous
Next

Required (1297) Label (1297) :
2-Column Wide Field 1297
Required (2297) Label (2297) :
2-Column Wide Field (2297)
FormSection
FormSection
FormSection

Account Competitors (1301): Account Competitors (1301)
Back Office Distribution Channel (1302): Back Office Distribution Channel (1302)
Account Condition (1303): Account Condition (1303)
Required 1304 Label (1304) :
Field (1304)

Account Role (1801): Account Role (1801)
Back Office Sales Organization (1802): Back Office Sales Organization (1802)
Account Status (1803): Account Status (1803)
Required (1804) Label (1804) :
Field (1804)

Algorithm Type (2301): Algorithm Type (2301)
BO Export Status (2302): BO Export Status (2302)
Alias (2303): Alias (2303)
Required (2304) Label (2304) :
Field (2304)

Figure 12. Simplified Model Form Applet

In Figure 12 on page 275, the underlying Web template in the figure has been modified to show the identifiers of the <swe: control> tags used to position controls within the applet. Note that the first text box (Account Competitors) that has been mapped to a <swe: control> tag has an identifier of 1301. The textboxes below it have identifiers of 1302 (Back Office Distribution Channel) and 1303 (Account Condition), respectively. The <swe:control> tag with the identifier 1304 does not have any control mapped to it. The adjacent columns of controls have identifiers that begin with 1801 (Account Role) and 2301 (Algorithm Type). In all, there are nine text boxes that have been mapped in this model form applet. Table 25shows the applet Web template item identifiers and the implied placement sequence of the identifiers for the text boxes in Figure 12 on page 275.

Table 25. Applet Web Template Identifiers from Figure 12

Applet Web Template Item Identifier	Implied Placement Sequence
1301	1
1302	2
1303	3
1801	4

Table 25. Applet Web Template Identifiers from Figure 12

Applet Web Template Item Identifier	Implied Placement Sequence
1802	5
1803	6
2301	7
2302	8
2303	9

The numeric order of the identifiers forms a logical placement sequence that will be correlated with that of the applet being migrated. The wizards only map controls from the applet being migrated to placeholders that have controls in the model applet.

The model applet also contains a number of buttons referenced in Figure 12 on page 275 by the circled letters D through I. These buttons will be automatically propagated to the migrated applets.

NOTE: In this example, the controls have names that correspond to fields on the Account business component. However, the controls could correspond to fields from any business component and it would not change how the wizards use them. The wizards do not care about names or fields of the controls in the model applet; they only evaluate which <swe:control> tags (placeholders) have controls mapped to them and the numeric value of their identifiers.

Figure 13 shows the results of using the model applet shown in Figure 12 on page 275 to migrate the ACME Contact Form Applet.

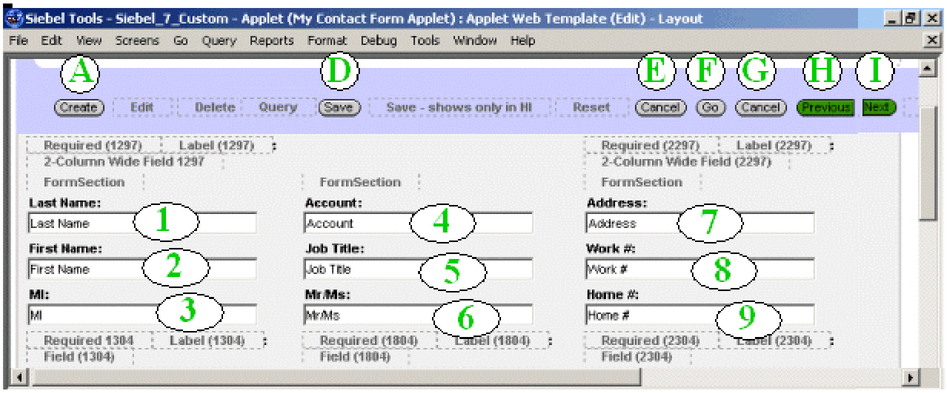


Figure 13. Migrated Acme Contact Form Applet

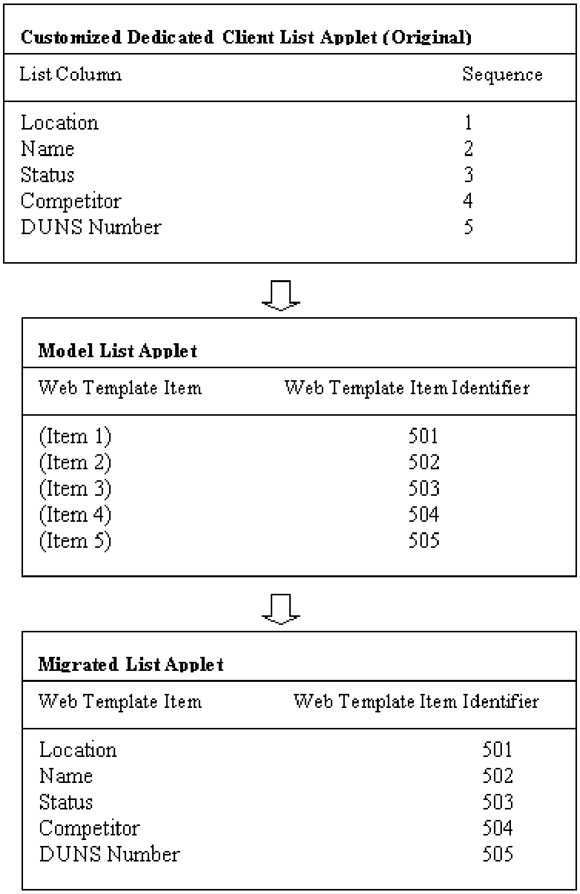
The circled numbers in Figure 13 correspond to the placement sequence of controls in the pre-migrated applet. The positioning of the textboxes in the migrated applet preserves the general orientation of controls in the original applet where controls with the lowest sequence number appear near the upper left corner and those with the highest appear near the lower right corner.

Figure 13 on page 277 also shows that controls 10 through 13 were not migrated at all; only nine textboxes were migrated because the model applet only had textboxes mapped to nine underlying placeholder tags. To avoid the problems that this can cause, you should be sure to use model applets that have model controls mapped to all the placeholders that you expect your migrated applets to use. This means that model applets tend to have controls mapped to most of the placeholders that are contained by the underlying Web template. However, there are some placeholders, such as those used for 2-Column Wide Fields (for example, placeholders 1297 and 2297 in Figure 12 on page 275 and Figure 13 on page 277) that may be better suited for usage on a case-by-case basis than as a general purpose anchor for controls.

All of the buttons on the migrated ACME Contact form applet, (Figure 13 on page 277), come from the model applet,(Figure 12 on page 275). The migrated applet contains buttons D through I, appearing the same as they did in the model form applet (Figure 12 on page 275); however, the caption property of Button A has changed from “New” to “Create.” The original ACME Contact form applet contains a control named “NewRecord” (button A) as does the model applet. Since it appears in the model applet, the wizard exposes it in the migrated applet. However, its caption property: “Create” reverts back to the value originally used by the ACME Contact form applet. By contrast, buttons B and C from the original applet are not exposed in the migrated applet because these controls have no corollary in the model.

The transformation between Figure 11 on page 274 and Figure 13 on page 277 also illustrates the results of merging labels and fields, a step that is discussed in “Preparing the Prior Customer Repository for the Merge” on page 160. This step automatically inactivates label controls and writes their caption property to the caption property of the adjacent text box controls. The net effect is that in Release 7.x, a single text box control is responsible for rendering itself as well as the label that appears above it. This simplifies the task of maintaining the controls in an applet and reduces the number of controls required to support the user interface.

The wizards behave similarly when migrating list applets, except that the placement sequence of list columns in the original list applet is determined by the list column sequence property instead of the top and left properties of text box controls. Figure 14 illustrates the rules for migrating list applets:



Customized Dedicated Client List Applet (Original)

List Column	Sequence
Location	1
Name	2
Status	3
Competitor	4
DUNS Number	5

Model List Applet

Web Template Item	Web Template Item Identifier
(Item 1)	501
(Item 2)	502
(Item 3)	503
(Item 4)	504
(Item 5)	505

Migrated List Applet

Web Template Item	Web Template Item Identifier
Location	501
Name	502
Status	503
Competitor	504
DUNS Number	505

Figure 14. Migration of a Dedicated Client List Applet to a Web Client List Applet

Migrating Customizations Using the Web Client Migration Wizard

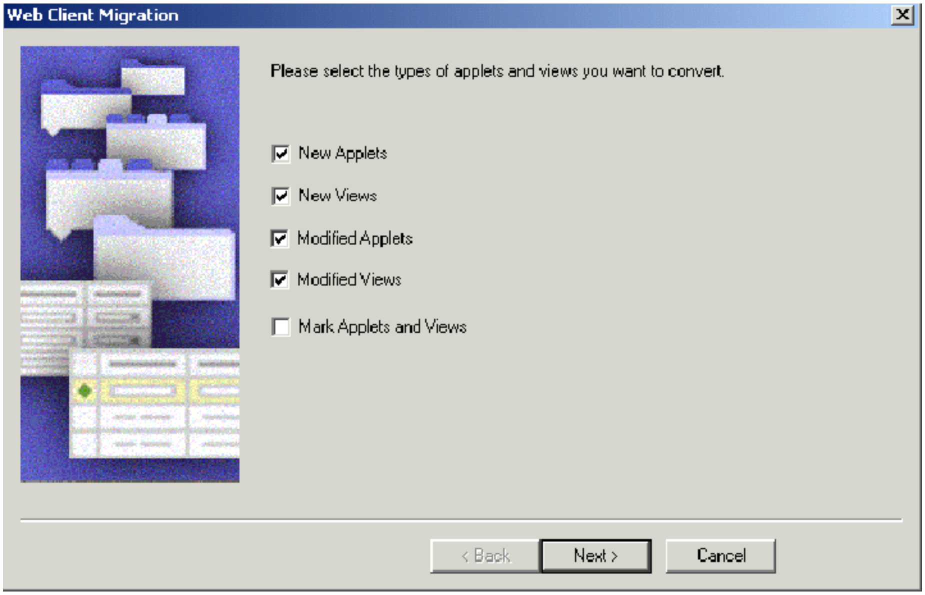
The following section describes how to use the Web Client Migration Wizard to migrate your customized and new applets and views.

NOTE: For information about using the Web Layout Wizard, see *Siebel Tools Reference*.

To migrate customizations to applets and views

1. Once the Application Upgrader has completed, go to the Application Upgrade Object List View in Siebel Tools. In Siebel Tools, select Screens > Application Upgrader > Application Upgrade Object List.

2. Invoke the Web Client Migration Wizard. In Siebel Tools select Tools > Upgrade > Web Client Migration.



Please select the types of applets and views you want to convert.

☑ New Applets

☑ New Views

☑ Modified Applets

☑ Modified Views

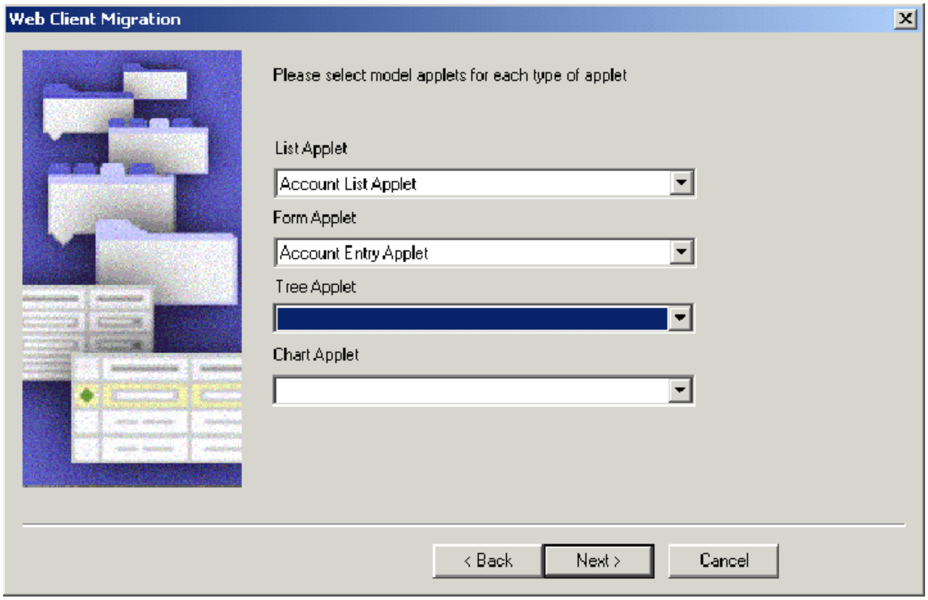
□ Mark Applets and Views

3 Select the various types of customizations that you wish to migrate:

- New applets
- New views
- Modified applets
- Modified views

4 At this point, you can choose to either have the wizard perform the migration or just mark the applets and views to be migrated, and then individually migrate the applets and views later.

5 On the next screen of the wizard, select model applets for each major applet class—Form, List, Chart, Tree.



Web Client Migration

Please select model applets for each type of applet

List Applet: Account List Applet

Form Applet: Account Entry Applet

Tree Applet:

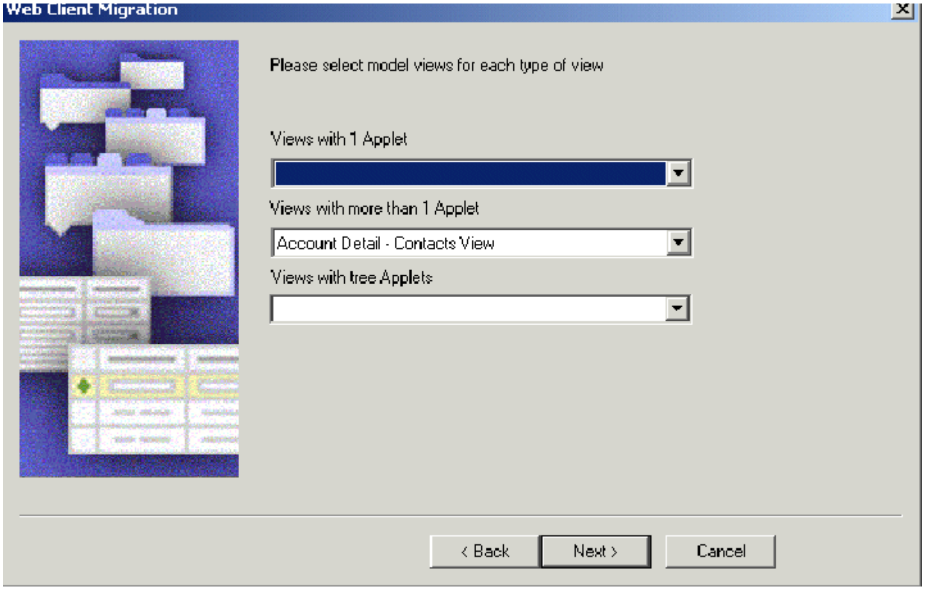
Chart Applet:

6 The following screen allows you to choose a model view for each category of view:

- View with one applet

- View with two or more applets
- Views containing tree applets

Any view of the appropriate category may be used as a model view. The information used by the wizard includes the templates to be used and the applet modes to use for the View Web Template Items.



Please select model views for each type of view

Views with 1 Applet

Views with more than 1 Applet: Account Detail - Contacts View

Views with tree Applets

7 Run the wizard.

When the wizard has completed, all the applets and views that have been migrated or are to be migrated are indicated. The Comments Field is used for this purpose and the following text is appended to the existing comments:

- `MigN.` This is appended to the Comments Field of new applets or views that have been migrated
- `MigM`. This is appended to the Comments Field of modified applets or views that have been migrated

- TBMN. This is appended to the Comments Field of new applets or views to be migrated
- TBMM. This is appended to the Comments Fields of modified applets or views to be migrated

Reviewing Applet Migration

After you have migrated applets, you should review them for accuracy.

To review new applet migration

1. In Siebel Tools, go to the Applet List View.
2. Search for applets that are marked as “MigN.”
3. Sort by Class.
4. For each applet in a given class, right-click on the applet. Select Edit Web Layout.
5. Review the layout, and make any appropriate changes to the layout using the Applet Web Editor.
6. If you notice that the same changes need to be made for a large number of applets, you can run the Web Layout Wizard to remigrate the applets.

   Common problems that may require remigration:

   - A set of non-Field controls is misplaced or absent in the Web Layout
   - The Template needs to be changed
   - Field controls are misplaced in the Web layout

   NOTE: Changes that do not fall into any of the above categories will need to be implemented manually for each applet.

Remigrating Applets

If a number of applets need to be remigrated, you can run the Web Layout Wizard to remigrate the applets.

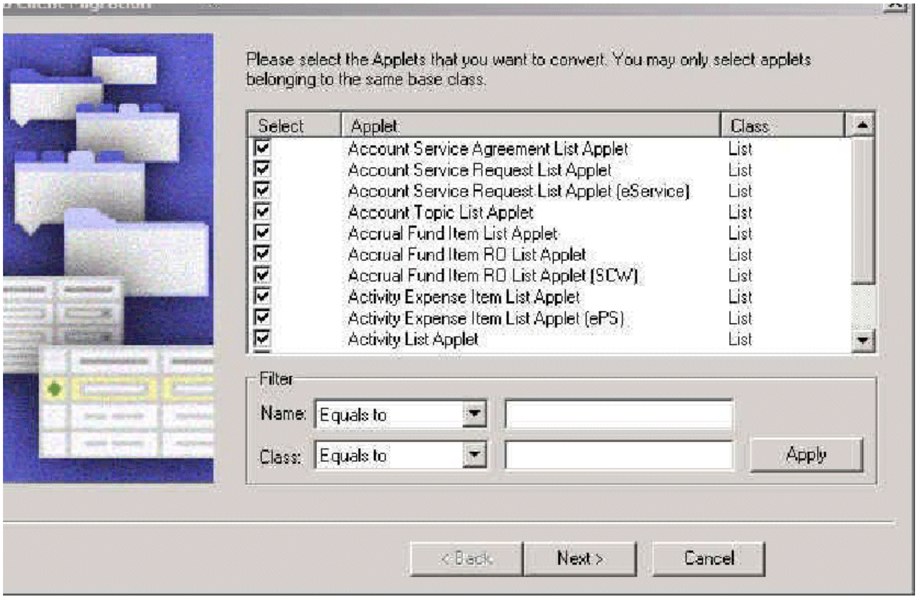
To remigrate applets

1 Before invoking the Web Layout Wizard, identify a model applet that contains the desired template and layout.

NOTE: You may need to modify an applet to obtain the desired characteristics.

2 Select the applets you want to remigrate using the selected model applet. You must select more than one applet.

3 Launch the Web Layout Wizard, by right-clicking and selecting Web Layout Wizard.



Please select the Applets that you want to convert. You may only select applets belonging to the same base class.

Select	Applet	Class
☑	Account Service Agreement List Applet	List
☑	Account Service Request List Applet	List
☑	Account Service Request List Applet (eService)	List
☑	Account Topic List Applet	List
☑	Accrual Fund Item List Applet	List
☑	Accrual Fund Item RO List Applet	List
☑	Accrual Fund Item RO List Applet (SCW)	List
☑	Activity Expense Item List Applet	List
☑	Activity Expense Item List Applet (ePS)	List
☑	Activity List Applet	List

Filter

Name: Equals to

Class: Equals to

4 Select the applets from the Applet List.

NOTE: You can only migrate one type of applet during each invocation. The wizard will not let you select applets of different types during a single invocation.

5 Select the model applet you identified in Step 1 on page 284.

For list applets, you can also specify the maximum number of List Columns that are to be initially visible. The remainder are available from the “Columns Displayed” dialog at run-time.

6 Convert the applets.

7 Review the conversion results for the set of applets converted in Step 6. If you are satisfied, go to Step 8. Otherwise, return to Step 6.

8 Repeat Step 4 on page 283 through Step 6 on page 285 for each Class of applet.

Reviewing Modified Applet Migration

After you completing remigration, you should review the remigrated applets.

To review modified applet migration

1 In Siebel Tools, go to the Applet List View.

2 Search for applets that are marked as “MigM.”

3 Sort by Class.

4 For each applet in a given Class, right-click on the applet and select Edit Web Layout.

5 Review the layout and make any appropriate changes to the layout using the Applet Web Editor.

6 Repeat until you have reviewed all applets.

Reviewing View Migration

After you have migrated views, you should review them to make sure that their Web Layout is accurate.

To review new view migration

1 In Siebel Tools, go to the View List view.

2 Search for views that are marked as “MigN.”

3 For each view of a given type, right-click on the view and select Edit Web Layout.

4 Review the layout and make any appropriate changes to the layout using the View Web Editor.

5 If you notice that certain changes need to be made for a large number of views, you can remigrate the views using the Web Layout Wizard.

Common problems that may require remigration:

- The mode of a View Web Template Item is incorrect.
- The Template needs to be changed.

NOTE: Changes that do not fall into any of the above categories need to be implemented manually for each view.

Remigrating Views

If you find that a number of views have not been properly migrated, you need to remigrate them using the Web Layout Wizard.

To remigrate views

1 Before invoking the Web Layout Wizard for a set of views, identify a model view that contains the desired template and layout.

NOTE: You may need to modify a view to obtain the desired characteristics.

2 Select all the views to be remigrated and launch the Web Layout Wizard in Siebel Tools by right-clicking and selecting Web Layout Wizard.

3 Select all the views that you wish to remigrate using the selected model view.

NOTE: You can not select views of different types during a single invocation. The wizard will not let you select views of different types during a single invocation.

4 Select the model view identified in Step 1 on page 286.

5 Convert the views.

6 Review the conversion results for the set of views converted in Step 5. If you are satisfied, go to Step 7. Otherwise, return to Step 1 on page 286.

7 Repeat Step 4 on page 286 through Step 5 for each type of view.

Reviewing Modified View Migration

After you have remigrated views, you need to review them for accuracy.

To review modified view migration

1 In Siebel Tools, go to the Views List view.

2 Search for views that are marked as “MigM.”

3 For each view, right-click on the view and select Edit Web Layout.

4 Review the layout. Make any appropriate changes to the layout using the View Web Editor.

5 Repeat until you have reviewed all remigrated views.

Migration of Scripts

The following sections describe in detail how to analyze scripts that need to be modified during the migration, and the process of making these modifications. This section first discusses scripts attached to UI objects—applets and controls. The final section discusses Business Component, Business Service, and Application scripts that reference UI methods.

Controls

In Release 7.x, the ControlName_Click event handler has been replaced by the "Method Invoked" property on Control objects. By specifying this property, when a user clicks on a control (for example, a Button), the client framework will check to see if the method has been implemented in Browser Script associated with the Applet_PreInvokeMethod event. If the method has not been implemented in Browser Script, the request will be sent to the server for execution. After this request is on the server, the Object Manager will check to see if the method has been implemented in the WebApplet_PreInvokeMethod event and then the BusinessService_PreInvokeMethod event. If the method has not been implemented in either Browser or Server script, an error will be raised.

In addition to the obsolescence of the ControlName_Click event handler, control methods that were formerly available in Siebel VB or eScript are no longer available. In Release 7.x, all control methods are available through Browser Scripting and will execute in the High Interactivity applications only. Table 26 provides a list of the control methods that are supported by Browser Script. For more information, see *Siebel Tools Online Help*.

Table 26. Control Method Syntax Summary

Method	Description	Syntax

Applet	Returns the parent applet for the control.	var oControl;
		var oApplet;
		oApplet = oControl.Applet();

Buscomp	Returns the corresponding business component for the control.	var oControl;
		var busComp;
		busComp = oControl.Buscomp();

GetValue	Returns the value of a control.	var oControl;
		var sValue;
		sValue = oControl.GetValue();

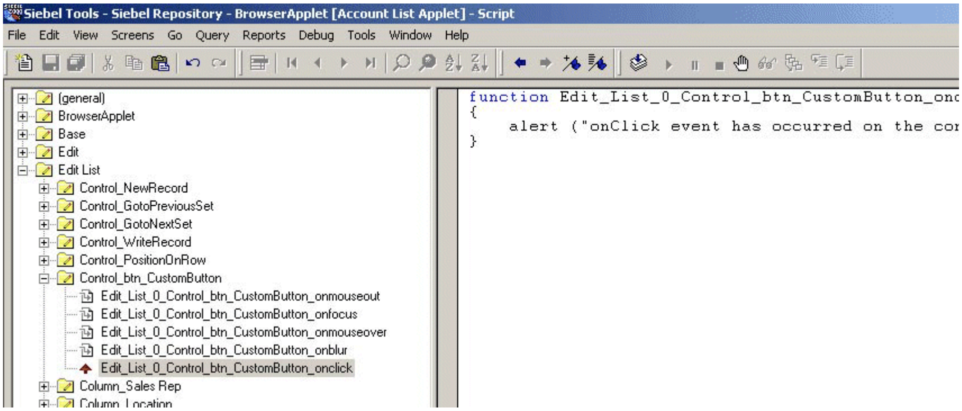
Table 26. Control Method Syntax Summary

Method	Description	Syntax

Name	Returns the name of the control.	var oControl;
		var sName;
		sName = oControl.Name();

SetValue	Sets the contents of the control to the indicated value.	var oControl;
		oControl.SetValue(*value*);

In addition to specifying the "Method Invoked" property, controls may also be scripted based upon native DOM events supported by the Browser. Figure 15 depicts the DOM events that may be scripted when a MiniButton control is added to an applet. The scripting of control-based DOM events is supported by High and Standard Interactivity applications and must be implemented in JavaScript.


Siebel Tools - Siebel Repository - BrowserApplet [Account List Applet] - Script
File
Edit
View
Screens
Go
Query
Reports
Debug
Tools
Window
Help
(general)
BrowserApplet
Base
Edit
Edit List
Control_NewRecord
Control_GotoPreviousSet
Control_GotoNextSet
Control_WriteRecord
Control_PositionOnRow
Control_btn_CustomButton
Edit_List_0_Control_btn_CustomButton_onmouseout
Edit_List_0_Control_btn_CustomButton_onfocus
Edit_List_0_Control_btn_CustomButton_onmouseover
Edit_List_0_Control_btn_CustomButton_onblur
Edit_List_0_Control_btn_CustomButton_onclick
Column_Sales Rep
Column_Location

```
function Edit_List_0_Control_btn_CustomButton_onc
{
    alert ("onClick event has occurred on the cor
}
```

Figure 15. DOM Events When MiniButton Control Added to Applet

Applets

As shown in Figure 16 on page 291, Applet Scripts implemented in Siebel 6.x will need to be moved to one or more of the following Browser or Server Script events:

- (Web) Applet Server Script

  - PreInvokeMethod
  - InvokeMethod
  - ShowControl
  - ShowListColumn
  - PreCanInvoke
  - Load
- Applet Browser Script
  - PreInvokeMethod
  - InvokeMethod
  - ChangeRecord
  - ChangeFieldValue
- Application Server Script
  - PreNavigate

- Navigate

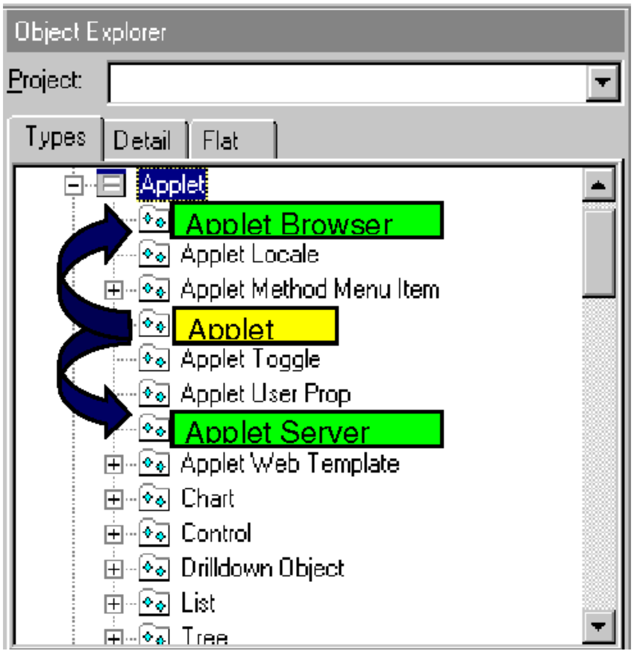


Figure 16. Movement of Applet Scripts

NOTE: In Release 7.x, the WebApplet_ShowControl and WebApplet_ShowListColumn event handlers are only supported for Standard Interactivity applications.

In many cases, the migration of Applet scripts will be relatively straightforward as several WebApplet events correspond to Siebel 6.x Applet events. Please note that though corresponding Browser or Server script event handlers may be available, the scripts will not be automatically migrated. Table 27 depicts Siebel 6.x events and the corresponding Release 7.x events available in either Browser or Server script.

Table 27. Siebel 6.x Applet Events and Release 7.x Equivalents

Siebel 6.x Applet Event	Release 7.x Equivalent
Applet_Load	WebApplet_Load (Server Event)
Applet_GotFocus	WebApplet_Load (Server Event)
Applet_ChangeFieldValue	Applet_ChangeFieldValue (Browser Event)
Applet_ChangeRecord	Applet_ChangeRecord (Browser Event)
Applet_PreInvokeMethod	WebApplet_PreInvokeMethod (Browser or Server Event)
Applet_InvokeMethod	WebApplet_InvokeMethod (Browser or Server Event)

In cases where there is no direct correlation between Siebel 6.x events and Release 7.x events (for example, Applet_LostFocus event), the functionality of the Siebel 6.x script will need to be evaluated and reimplemented. Depending upon the functionality provided by the script, scripts on the Applet_LostFocus event could be reimplemented as a Server script and implemented on the TheApplication.Navigate or PreNavigate event.

Some applet level methods are no longer available in Release 7.x. The obsolete Applet Methods are listed in Table 28. In some cases, the methods have been reimplemented as Browser Script Methods, as in the case of FindControl and FindActiveX Control. Other methods, such as PostChanges have been replaced by properties that can be configured in Siebel Tools (Immediate Post Changes property on a Field), and a few others will require modifications to scripts that use these methods.

Table 28. Obsolete Applet Methods

Applet Method
ActiveControl
FindControl

Table 28. Obsolete Applet Methods

Applet Method
FindActiveXControl
GotoControl
PostChanges
Drilldown

Business Components, Business Services, and Application Scripts

The majority of Application, Business Components, and Business Services scripts implemented in Siebel 6.x can remain unchanged and should not require any modification after the upgrade. In instances where existing scripts reference methods that have been made obsolete or contain references to the Siebel user interface, the script will need to be edited. Table 29 lists the Application and Business Component methods that have been made obsolete in Release 7.x.

Table 29. Obsolete Business Components, Business Services, and Application Scripts

Obsolete Business Components, Business Service, and Application Scripts
Application.MsgBox
Application.InputBox
Application.ActiveBusComp
Application.ActiveApplet
Application.GotoApplet
Application.ShowStatus
Application.ActiveControl
Application.FindApplet
BusComp.AllowNotification
BusComp.SuppressNotification

In many cases, implementation alternatives are available in Release 7.x that offer comparable functionality to the obsolete methods. For example, the ActiveBusComp method can typically be replaced with a combination of ActiveBusObject and GetBusComp. By using these two methods together in a script, a handle to the active business component can be obtained. Additionally, MsgBox and InputBox methods can typically be replaced with some core functions available in JavaScript (through Browser script) and some new methods introduced in Release 7.x. For example, using Browser Script, you can use code JavaScript features to provide warnings, dialog boxes, or input boxes through the alert, confirm, and prompt functions. For more information about using JavaScript features available in Browser Script to replace MsgBox and InputBox, see FAQ 1562 on Siebel SupportWeb.

In order to provide an alert notification from Server Script, the RaiseError and RaiseErrorText methods were introduced (for detailed information on RaiseError and RaiseErrorText please refer to *Siebel Object Interfaces Reference* and *Siebel Tools Online Help*). These methods allow an error message to be displayed as an alert in the Web Client, though it should be noted that any lines of code that follow the RaiseError or RaiseErrorText methods will not be executed. For more information about using RaiseError and RaiseErrorText methods, as well as additional information about using Browser Script and Server Script in Release 7.x, see *Siebel Object Interfaces Reference* and Technical Note 386 on Siebel SupportWeb.

Migration of Client-Side Interfaces

This section describes the migration of client-side interfaces.

Outbound COM Interfaces

Client side application integration in the zero-footprint Web client can be accomplished by using two methods provided by Jscript.Net, Microsoft’s implementation of the ECMA 262 language. These methods are:

ActiveXObject – Enables and returns a reference to an Automation object

GetObject – Returns a reference to an Automation object from a file

Using the ActiveXObject function for example, you can access properties and methods of Excel, including the Application object and the ActiveSheet. Cells collection as shown in the example below.

```
var ExcelApp;
var Sheet;

ExcelApp = new ActiveXObject("Excel.Application");

Sheet = new ActiveXObject("Excel.Sheet");

// Make Excel visible

Sheet.Application.Visible = true;

// Place some text in the first cell

Sheet.ActiveSheet.Cells(1,1).Value = "This is row 1 column A";

// Save the sheet.

Sheet.SaveAs("C:\\TEST.XLS");

// Close Excel and quit.

Sheet.Application.Quit();
```

NOTE: The ActiveXObject and GetObject methods can only be used with Internet Explorer 3.0 or greater. For additional information on the ActiveXObject and GetObject functions, please refer to the Microsoft Web site (http://www.microsoft.com) and the Jscript.Net documentation.

If you are deploying the Mobile Web Client, you can use native functions provided by Siebel VB or eScript to handle client side application integration, in addition to the ActiveXObject and GetObject functions described above. The native Siebel VB and eScript functions commonly used for client-side application integration are:

- Siebel VB
  - CreateObject (to access a client side Automation object)
  - Declare (to access an external DLL)
- eScript
  - COMCreateObject (to access a client side Automation object)
  - CORBACreateObject (to access a remote CORBA interface)

- SELib.DynamicLink (to access an external DLL)

For additional information on the Siebel VB and eScript functions listed above, please refer to *Siebel Object Interfaces Reference* and *Siebel Tools Online Help*.

Inbound COM Interfaces

In Release 7.x, external applications can interact with the Siebel Web Client through the Mobile Web Client Automation Server and Web Client Automation Server. The Mobile Web Client Automation Server is similar in functionality to the COM Automation Server available in Siebel 6.x, though it does not expose applet or control objects.

In order to allow external applications to interact with the zero-footprint Web client, Release 7.x introduces the Web Client Automation Server. The Web Client Automation Server allows external applications to invoke Business Services and manipulate Property Sets from external, COM based applications. The Web Client Automation Server is implemented as a small COM object resident within the browser (IE 5.0 or greater). The Web Client Automation Server is supported with the High Interactivity applications only.

To enable the Web Client Automation Server, in the [SWE] section of your application's configuration file you will need to make sure that the "EnableWebClientAutomation" parameter is set to TRUE. With this parameter set to TRUE, a small ActiveX Control will be downloaded to the desktop and the SiebelHTMLApplication process will be started. This process will terminate when the Siebel Web Client is gracefully terminated. Please note the modification Browser security settings may be required to allow use of the Web Client Automation Server.

Below is an example of a small Microsoft Visual Basic application that uses the Web Client Automation Server and the EAI Siebel Adapter business service to query for a list of Accounts that start with the letter A and displays the result in a message box.

```
Private Sub Command1_Click()

    Dim siebelApp As SiebelHTMLApplication

   Dim inPs As SiebelPropertySet, siebelMsg As SiebelPropertySet

    Dim listOfAcct As SiebelPropertySet, acctPs As
SiebelPropertySet

    Dim outputPs As SiebelPropertySet

    Dim svc As SiebelService
    Dim i As Long, j As Long, iPos As Long

    Dim acctList As String

    Dim status As Boolean

    Set siebelApp = GetObject("",
"SiebelHTML.SiebelHTMLApplication.1")

    Set inPs = siebelApp.NewPropertySet

    Set siebelMsg = siebelApp.NewPropertySet

    Set listOfAcct = siebelApp.NewPropertySet

    Set acctPs = siebelApp.NewPropertySet

    Set outputPs = siebelApp.NewPropertySet

    Set svc = siebelApp.GetService("EAI Siebel Adapter")



    ‘ The following lines will construct a property set to query
on the Sample Account

    ‘ Integration Object

    siebelMsg.SetType ("SiebelMessage")

    status = siebelMsg.SetProperty("IntObjectName", "Sample
Account")

    status = siebelMsg.SetProperty("MessageId", "")

    status = siebelMsg.SetProperty("Message Type", "")



    listOfAcct.SetType ("ListOfSample Account")



    acctPs.SetType ("Account")

    status = acctPs.SetProperty("Name", "A*")
    iPos = listOfAcct.AddChild(acctPs)

    iPos = siebelMsg.AddChild(listOfAcct)

    iPos = inPs.AddChild(siebelMsg)


    ‘ Now that the Integration Object has been constructed, query
for Accounts starting with A

    status = svc.InvokeMethod("Query", inPs, outputPs)


    If status then

    j = 0

    i =
outputPs.GetChild(0).GetChild(0).GetChild(0).GetChild(0).GetProp
ertyCount


    Do While j < i

        acctList = acctList &
outputPs.GetChild(0).GetChild(0).GetChild(0). _

            GetChild(j).GetProperty("Name") & Chr(10) & Chr(13)

        j = j + 1

    Loop

    MsgBox (acctList)

        End If

End Sub
```

NOTE: For additional information on the Mobile Web Client Automation Server or the Web Client Automation Server please refer to *Siebel Tools Online Help*.

Continuing with the Upgrade

To continue with the upgrade, follow the instructions below for the environment you are upgrading:

- **Development Environments.** If you have completed migrating the client configuration to the Web Client, go to Chapter 3, “Preupgrade Tasks,” to begin preugpgrade tasks for the production environment.
- **Production Environments.** If you have completed postupgrade tasks for the production environment and migration of the client configuration to the Web Client, you have completed your upgrade.

Upgrade Planning Worksheet A

This appendix provides a sample worksheet for planning an upgrade.

Worksheet

Table 30 is a sample worksheet for planning an upgrade.

Table 30. Worksheet for Upgrade Planning

Information Needed	Value at Your Site	Comment
Siebel tableowner account username and password		The database account that owns the Siebel eBusiness Applications database objects. For DB2 and Oracle, `SIEBEL` is the default username and password.
Siebel Administrator username and password		This must be a valid RDBMS username and password, and it must be set up as a Siebel Systems employee. The employee record must have the “Siebel Administrator” responsibility. `SADMIN` is the default administrator username and password.
ODBC data source name		This is the name of the ODBC data source used to connect to the Siebel eBusiness Applications database server. The default is *"SAS_enterprise_server_siebel_server".*
Application Server root directory		This is the root directory of your existing Siebel Server software installation.
File System directory name		This is the path from the Siebel Server to the Siebel File System, which may use a share name, drive mapping, or other method, depending on your Siebel File System.
Siebel Data Segment Name		This is the name of the tablespace or segment on which the Siebel eBusiness Applications tables are stored in Oracleand DB2 UDB installations.
Siebel Index Segment Name		This is the name of the tablespace or segment on which the Siebel eBusiness Applications indexes are stored in Oracle and DB2 UDB installations.

Upgrading Without a Development Environment

B

This appendix outlines the tasks involved in upgrading your Siebel application if you have not customized your application and therefore do not have a development environment. If you are upgrading without a development environment, you will upgrade a test environment and then upgrade your production environment using the procedures described in "Upgrading Without a Development Environment" on page 304.

The process for upgrading without a development is illustrated in Figure 17.

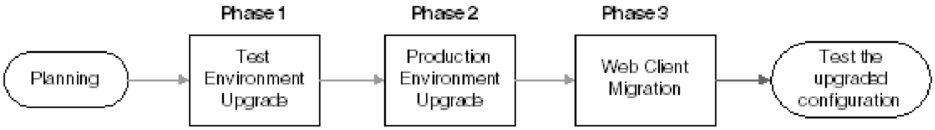


Figure 17. Process for Upgrading without a Development Environment

Upgrading Without a Development Environment

If you are upgrading without a development environment, upgrade a test environment (using a copy of your production database) before you upgrade your production environment. After you have verified that the upgrade was successful in a test environment, you can safely upgrade your production environment.

Complete the following steps to upgrade your production repository and database schema.

CAUTION: This procedure will overwrite all customizations. Do not attempt this procedure if you have a development environment or if you have customizations or configuration changes to your Siebel deployment.

To upgrade without a development environment

1 Upgrade your test environment. Navigate to `/dbsrvr/common` and locate the mstrep.dat file. Copy the `mstrep.dat` file and rename it `custrep.dat`. Place the `custrep.dat` file in the `/dbsrvr/platform` directory.

2 Use the configuration utility to upgrade the Siebel database schema on your test environment, as instructed in "Upgrading the Siebel Database Schema" on page 147.

   a Specify parameters you listed in the Upgrade Planning Worksheet.

   b Set the following parameter as shown:

   ```
   Environment Type = "Production"
   ```

   NOTE: You might need to change the ODBC data source to point to your test database before upgrading the Siebel database schema (upgrep) and the custom database schema (upgphys).

3 Use the configuration utility to upgrade the custom database schema on your test environment, as instructed in "Upgrading the Custom Database Schema" on page 189.

   - Specify parameters you listed in the Upgrade Planning Worksheet.

4 Review the upgrade log files to verify that no unacceptable errors are reported. See "Reviewing the Upgrade Log Files" on page 153.

> NOTE: You should save log files from the SiebSrvr\Log directory into a Zip file so that you can review them later. If you do not save them, they will be moved to the `SiebSrvr\Log` archive directory and will eventually be removed when the machine is rebooted or Siebel Server services are restarted.

If you find errors that are not included in the `errors.rtf` file in the *`DBSRVR_PLTFRM_ROOT`* directory, contact Siebel Technical Support.

> CAUTION: Before you begin the upgrade of your production environment, verify that there are no repositories named "Siebel Repository" or "New Customer Repository" in your database.

5 Upgrade your production environment. Use the configuration utility to upgrade the Siebel database schema on your production environment, as instructed in "Upgrading the Siebel Database Schema" on page 147.

a Specify parameters you listed in the Upgrade Planning Worksheet.

b Make sure that the following parameter is set as shown below:

```
Environment Type ="Production"
```

> NOTE: You might need to change the ODBC data source to point to your production database before upgrading the Siebel database schema (upgrep) and the custom database schema (upgphys).

6 Use the configuration utility to upgrade the custom database schema on your production environment, as instructed in "Upgrading the Custom Database Schema" on page 189.

- Specify parameters you listed in the Upgrade Planning Worksheet.

7 Verify that no unacceptable errors are reported in the error logs. See "Reviewing the Upgrade Log Files" on page 153.

Upgrading Without a Development Environment

Error Messages C

This appendix lists and defines error messages you may encounter when you perform certain upgrade procedures.

Error Messages for Database Server Configuration Utility

Table 31 describes errors that may occur during upgrade of your Siebel Database Schema (upgrep) or custom database schema (upgphys).

Table 31. Upgrade Errors

Error Message	Acceptable?	Description
`Error ORA-24345 "A Truncation or null fetch error occurred".`		This is a well-known issue and TECHNICAL NOTE 0345 is published on support web.

`Failure: Dictionary for EIM_AGREE_ITEM different from physical schema`
`Dictionary column X_LINE_TYPE not in physical schema!`
	No	Upgrade reports this error for custom extension columns which are not upgraded to the physical schema but are present in the logical layer.
		Fix this by running the DB Server Configuration utility with Synchronize schema definition.

Error Messages for Repository Merge

Table 32 describes errors that may occur when you perform a repository merge.

Table 32. Repository Merge Errors

Error Message	Acceptable?	Description

`!!ERROR::CommitCopy Instance EXITING with error - The same values for 'Column Name' already exist.`
`!!ERROR::INSERTION FAILED. Type Workflow Policy Column, NamePath - Sales Stage Name`
	Yes	This error occurs because a workflow based on CURR_STG_ID was created in the previous version. The upgrade creates another workflow based on the same column to fix the defect and hence the error occurs.

`!!ERROR::REQUIRED FKey not found in 'New Customer Repository'`
`!!ERROR::Type - Workflow Policy Component Col, strNP - OpportunitySales StageSales Stage Name, fieldName - Workflow Column Id, fieldVal - 1-2G0-3F9R`
`!!ERROR::INSERTION FAILED. Type Workflow Policy Component Col, NamePath - OpportunitySales StageSales Stage Name`
	Yes	This error occurs because a workflow based on CURR_STG_ID was created in the previous version. The upgrade creates another workflow based on the same column to fix the defect and hence the error occurs.

Error Messages for DICTUTL

Table 33 describes errors that may occur in the dictutl.log file.

Table 33. DICTUTL Errors

Error Message	Acceptable?	Description
`Error: Col Rel 106(1-5CC-84WT) Target Col points to nonexistent column`	No	This message is related to Dock Object Visibility rule with Check Dock Object type. The Target Column in the Visibility rule points to a column that is invalid. To address this error, use Siebel Tools to configure the Dock Object Visibility rule so that it points to a valid column.

Index